

# **METODOLOGÍA PARA LA IMPLEMENTACIÓN DE PROCESOS WORKFLOW EN UNA COMPAÑÍA DE SEGUROS**

Realizado por:  
Erika María Gallego Lineros

Proyecto de grado para optar al título de ingeniero de sistemas

UNIVERSIDAD EAFIT

Departamento de Informática y Sistemas

MEDELLÍN

2006

# **METODOLOGÍA PARA LA IMPLEMENTACIÓN DE PROCESOS WORKFLOW EN UNA COMPAÑÍA DE SEGUROS**

Realizada por:

Erika María Gallego Lineros

Asesora principal:

Lina María Jaramillo Terán

Ingeniera de Sistemas

Especialista en Sistemas de Información

## **AGRADECIMIENTOS**

Agradezco a mi guía y asesora Lina María Jaramillo Terán, por su constante y desinteresado apoyo; su dedicación hizo que este trabajo fuera una fuente de conocimiento.

## CONTENIDO

LISTA DE FIGURAS .....	vii
LISTA DE TABLAS .....	ix
ABREVIATURAS .....	x
INTRODUCCIÓN .....	1
OBJETIVOS .....	4
PARTE I ORIGEN Y EVOLUCIÓN DE LA TECNOLOGÍA WORKFLOW .....	5
CAPITULO 1. MARCO TEÓRICO .....	5
1.1 ORÍGENES DE WORKFLOW .....	5
1.2 EVOLUCIÓN DE WORKFLOW .....	10
CAPITULO 2. SISTEMAS DE GESTIÓN DE FLUJOS DE TRABAJO .....	14
2.1 CLASIFICACIÓN DE LOS SISTEMAS DE GESTIÓN DE FLUJOS DE TRABAJO .....	16
2.2 LIMITACIONES DE LOS SISTEMAS DE GESTIÓN DE FLUJOS DE TRABAJO .....	20
2.3 ESTANDARIZACION DE LOS SGFT .....	22
2.3.1 Necesidad de estandarizar .....	22
2.3.2 Modelo de referencia de la WfMC .....	24
PARTE II PRELIMINARES PARA EL DESARROLLO DE UNA METODOLOGÍA DE AUTOMATIZACIÓN DE PROCESOS WORKFLOW .....	28
CAPITULO 3. ELEMENTOS QUE COMPONEN UN FLUJO DE TRABAJO .....	28
3.1 DIMENSIONES DE UN FLUJO DE TRABAJO .....	28
3.2 COMPONENTES DE UN FLUJO DE TRABAJO .....	30
3.2.1 Proceso .....	31
3.2.1.1 Actividad .....	34
3.2.1.2 Subproceso .....	35
3.2.1.3 Condiciones de Transición .....	36
3.2.1.4 Flujo de Control .....	40

3.2.2.1 Actores y Modelo Organizacional .....	41
3.2.2.2 Datos .....	44
3.2.3 Relación entre las actividades y los recursos.....	46
CAPITULO 4. PRINCIPALES METODOLOGÍAS QUE PERMITEN EL MODELAMIENTO WORKFLOW DENTRO DE LAS ORGANIZACIONES.....	48
4.1 QUÈ ES UNA METODOLOGÍA DE DESARROLLO DE SOFTWARE .....	48
4.2 METODOLOGÍAS PARA MODELAR PROCESOS WORKFLOW .....	49
4.2.1 Modelo ICN (Information Control Net) .....	49
4.2.2 Redes de Petri .....	51
4.2.3 Workflow Management Technology .....	52
4.2.3.1 Etapas de un Workflow .....	54
4.2.3.2 Interacciones de un Workflow.....	54
CAPITULO 5. MODELO GENÉRICO PARA LA AUTOMATIZACIÓN DE PROCESOS WORKFLOW.....	56
5.1 ENTORNO DE SOPORTE AL PROCESO DE DESARROLLO.....	62
PARTE III METODOLOGÍA PARA DESARROLLAR PROCESO WORKFLOW EN UNA COMPAÑÍA DE SEGUROS S.A.....	65
CAPITULO 6. EL PROCESO UNIFICADO DE DESARROLLO DE SOFTWARE .....	65
6.1 CARACTERÍSTICAS PRINCIPALES.....	66
6.2 FASES DEL PROCESO UNIFICADO.....	68
CAPITULO 7. TECNICAS UTILIZADAS PARA AUTOMATIZAR PROCESOS WORKFLOW. ....	69
7.1 DESCRIPCIÓN DEL SISTEMA ACTUAL .....	69
7.2 OBTENER INFORMACIÓN SOBRE EL DOMINIO DEL PROCESO A AUTOMATIZAR .....	69
7.3 OBJETIVOS DEL PROCESO A AUTOMATIZAR .....	70
7.4 ESPECIFICACIÓN DE PARTICIPANTES DEL PROYECTO .....	73
7.5 REPOSITORIO DE ARTEFACTOS Y DE VERSIONAMIENTO .....	75

7.6 MAPA DE PROCESOS .....	77
7.7 GLOSARIO DE TÉRMINOS .....	79
7.8 LEVANTAMIENTO DE REQUISITOS.....	79
7.9 IDENTIFICACIÓN DE LAS REGLAS DE NEGOCIO .....	80
7.10 UML COMO LENGUAJE PARA EL ANÁLISIS DE REQUISITOS .....	81
7.10.1 Levantamiento de requisitos con UML.....	82
7.10.1.1. Actores .....	87
7.10.2 Diagramas UML .....	89
7.10.2.1 Diagramas de casos de uso .....	89
7.10.2.2 Diagrama de actividades .....	91
7.10.2.3 Diagrama de Procesos .....	92
7.10.2.4 Diagramas de clases del dominio .....	94
7.10.2.5 Diagrama de componentes.....	94
7.10.2.6 Diagrama de secuencia .....	98
7.10.2.7 Diagrama de despliegue.....	99
7.10.3 Arquitectura.....	99
7.11 MATRIZ DE ENRUTAMIENTO.....	101
7.12 TIPOS DE DATOS Y DOCUMENTOS.....	105
7.13 ROLES.....	106
7.14 INTEGRACIONES .....	107
7.15 VALIDACIÓN DE REQUISITOS .....	108
CONCLUSIONES .....	109
TRABAJOS FUTUROS.....	110
BIBLIOGRAFIA.....	111

## LISTA DE FIGURAS

Figura 1. Cuadrante mágico de Gartner .....	10
Figura 2. Conceptos básicos y terminología Workflow .....	14
Figura 3. Clasificación de los SGFT según complejidad y estructura de las.....	17
actividades involucradas.....	17
Figura 4. Clasificación de los SGFT según la similitud de los procesos de .....	18
negocio y su valor en la Organización. ....	18
Figura 5. Estándares de la WPMC.....	24
Figura 6. Modelo de Referencia de un SGFT .....	25
Figura 7. Dimensiones de un flujo de trabajo.....	29
Figura 8. Componentes de un Flujo de Trabajo.....	30
Figura 9. Jerarquía de Generalización.....	32
Figura 10. Diagrama de transición de estados.....	33
Figura 11. Condiciones de bifurcación.....	36
Figura 12. Condiciones de unión .....	38
Figura 13. Jerarquía de Generalización para los recursos .....	41
Figura 14. Modelo organizacional.....	43
Figura 15. Relaciones entre los componentes del flujo de trabajo.....	47
Figura 16. Grafo de actividades ICN.....	50
Figura 17. Interacciones Dentro de Workflow .....	53
Figura 18. Actividades del ciclo de Vida en Workflow.....	56
Figura 19. Vista general del proceso unificado .....	68
Figura 24. Mapa de proceso modelado en papel.....	78
Figura 26. Diagrama de casos de uso .....	90
Figura 27. Representación gráfica de las relaciones de includes y extends.....	90
Figura 28. Diagrama de actividades .....	92
Figura 29. Diagrama de procesos.....	93
Figura 30. Diagrama de clases del dominio.....	94

Figura 31. Diagrama de componentes.....	95
Figura 32. Diagrama de secuencia .....	98
Figura 33. Modelo de arquitectura 4+1 .....	100



## LISTA DE TABLAS

Tabla 1. Secuencia de interacciones .....	55
Tabla 2. Interacciones de Workflow de Petición .....	55
Tabla 3. Entorno de soporte al proceso de desarrollo .....	64
Tabla 4. Líder por cada etapa de desarrollo del proceso a automatizar .....	<b>¡Error!</b>
<b>Marcador no definido.</b>	
Tabla 5. Plantilla de objetivos .....	70
Tabla 6. Plantilla de Roles y responsabilidades del equipo de trabajo .....	73
Tabla 7. Estructura de modificaciones .....	76
Tabla 8. Estándar de archivos .....	76
Tabla 9. Estándar de directorios .....	76
Tabla 10. Versiones de los flujos .....	77
Tabla 11. Estructura glosario de términos .....	79
Tabla 12. Plantilla para el levantamiento de requisitos .....	83
Tabla 13. Plantilla para la descripción de casos de uso extendidos .....	84
Tabla 14. Plantilla de Actores .....	88
Tabla 14. Ejemplo: Matriz de enrutamiento para el usuario .....	104
Tabla 15. Ejemplo: Matriz de enrutamiento para el analista .....	104
Tabla 16. Matriz de datos .....	106
Tabla 17. Plantilla de roles.....	107
Tabla 18. Plantilla integraciones de aplicaciones.....	107

## **ABREVIATURAS**

BPM: Business Process Modeling.

BPR: Business Process Reengineering.

HMP: Herramienta de Modelado de Procesos.

DFD: Data Flow Diagrams.

UML: Unified Modeling Language.

TI: Tecnología de Información.

SDI: Sistema de documentación informática.

REM: Requisite Management.

## INTRODUCCIÓN

Este proyecto de grado tiene como objetivo principal definir una metodología para la implementación de procesos Workflow en una Compañía de Seguros, realizando un trabajo investigativo sobre el origen y evolución de Workflow, identificando los conceptos mas importantes que se requieren para implementar la automatización de un proceso .

Con el ingreso de nuevos competidores en el sector financiero y asegurador al país, las amenazas para la Compañía de Seguros han aumentado.

Mediante las nuevas tendencias tecnológicas como es el caso de la tecnología Workflow, la cual automatiza, administra los procesos tanto de apoyo como el core del negocio, así las compañías de seguros, puede competir para adquirir nuevos negocios y mercados, ya que esta tecnología permite mejorar tiempos de respuesta, servicio al cliente y optimización del ciclo de los procesos, de tal manera que se logre mantener a la altura de las grandes aseguradoras internacionales.

Dentro de un proceso, la información tradicionalmente es manejada a través de cartas, fotos, documentos físicos, correos electrónicos, entre otros; dicha información presenta problemas tales como desperdicio de papel, la pérdida de información, gran cantidad de tiempo en el ciclo del proceso y redigitalización de documentos. Es así como la tecnología Workflow permite automatizar, coordinar, controlar y monitorear la secuencia de las actividades necesarias para realizar un proceso del negocio, logrando incrementar considerablemente la eficiencia.

Durante la implantación de la tecnología Workflow en la Compañía de Seguros y el desarrollo del primer proceso en la herramienta, se vio la necesidad de definir una metodología que permita desarrollar proyectos bajo este

nuevo reto que asume la Compañía. Actualmente la compañía cuenta con una metodología de desarrollo de productos software la cual no satisface completamente las necesidades para el desarrollo sobre esta nueva tecnología.

El primer paso en este proyecto fue investigar el origen de Workflow, cómo ha evolucionado a través del tiempo hasta llegar a las herramientas que actualmente nos ofrece el medio; para esto, fue necesario consultar bibliografía especializada. Se investigaron también los diferentes tipos de metodologías que existen en el medio para este desarrollo automatizado. De igual manera, se definió un modelo genérico el cual, cualquier compañía puede utilizar para la automatización de sus procesos, ya que abarca de una forma genérica los aspectos y conceptos mas relevantes del desarrollo o implementación de un proceso con tecnología Workflow.

Paralelamente se realizó un análisis y estudio de la metodología actual con la que cuenta la compañía para el desarrollo de Software, siendo la base para la metodología propuesta en este trabajo.

A continuación se presenta el contenido y los temas que se tocan en cada parte del trabajo.

Este trabajo está dividido en tres partes y estos a su vez en capítulos; la primera parte abarca el marco teórico de la tecnología Workflow como su origen y evolución hasta llegar a lo que hoy conocemos como Sistemas de gestión de flujos de trabajo, que son las herramientas que nos brinda el mercado para automatizar procesos Workflow. La segunda parte trata sobre los conceptos mas importantes y relevantes que se deben tener en cuenta para utilizar la tecnología Workflow, también se identifican metodologías que permiten modelar procesos y por último se expone un modelo genérico para la automatización de procesos en cualquier compañía. La tercera parte contiene una sección en donde se enuncian algunas

técnicas que son de gran ayuda durante el ciclo de vida del desarrollo del proceso a automatizar, esta sección cuenta con algunas plantillas que son de mucha utilidad para documentar y estructurar el análisis, desarrollo y mantenimiento del proceso.

Finalmente se presentan las conclusiones y trabajos futuros de este proyecto de grado, en cuanto a recomendaciones, vale la pena rescatar que la metodología para la automatización de procesos Workflow propuesta está sujeta a verificación y posibles mejoras ya sea por las lecciones aprendidas que las distintas aplicaciones generen o a evolucionar con las nuevas tendencias del mercado.

## OBJETIVOS

El objetivo principal de este trabajo es definir una metodología para implementar procesos Workflow en una Compañía de Seguros.

Este objetivo general se desglosa en los siguientes objetivos específicos:

- ⌚ Conocer el marco teórico de la tecnología Workflow, su origen y evolución.  
Identificar conceptos principales para la automatización de procesos Workflow.
- ⌚ Analizar e investigar metodologías existentes para la automatización de procesos Workflow.
- ⌚ Identificar técnicas que apoyen la implementación de procesos Workflow en una Compañía de Seguros.
- ⌚ Plantear un modelo genérico que permita identificar etapas y actividades a desarrollar en la implementación de un proceso Workflow en cualquier Compañía u organización.
- ⌚ Identificar la relación que existe entre los distintos componentes de los flujos de trabajo.
- ⌚ Identificar paso a paso las actividades a realizar en la implementación de un proceso Workflow, asignando un responsable por cada actividad.

## **PARTE I ORIGEN Y EVOLUCIÓN DE LA TECNOLOGÍA WORKFLOW**

### **CAPITULO 1. MARCO TEÓRICO**

#### **1.1 ORÍGENES DE WORKFLOW**

Recientes investigaciones aseguran que la tecnología Workflow se basa en la idea que algunas cosas son realizadas más efectivamente por las máquinas o computadores que por las personas. Los humanos somos buenos para tomar decisiones, innovar, identificar hechos inesperados, pero usualmente no somos eficientes en actividades como buscar un documento entre otros, tener presentes los vencimientos de las tareas que se tienen que realizar dentro de ciertos plazos, así como el asegurarse de que el trabajo terminado pase de un lugar a otro respetando el procedimiento definido (secuencia).

En los años 70's, 80's y parte de los 90's, el principal uso de los ordenadores en las organizaciones era la automatización de las actividades individuales. Actualmente la situación es diferente ya que en los últimos años las dependencias o subdivisiones de las organizaciones son vistas como una sola, es decir, no se habla de actividades por áreas como actividades individuales, sino cómo se integran entre ellas. En el campo o área del desarrollo de software existen métodos y notaciones de análisis y diseño consolidados como UML (Unified Modeling Language) [Booc99], herramientas de desarrollo cada vez más potentes como generadores de código y modelos de ciclo de vida basados en el uso de las mismas. Todo ello ha motivado que se esté produciendo un cambio de orientación hacia los procesos organizacionales o procesos de negocio. El interés de las

organizaciones no está limitado únicamente al desarrollo de software que automatice ciertas actividades individuales, sino que su objetivo final es la automatización de todo el proceso de negocio, ya que de ello depende en gran parte su competitividad. Por esto, surgen nuevas necesidades de capturar, modelar, ejecutar y monitorear los procesos los cuales son vistos como un conjunto de procedimientos o actividades relacionadas, para alcanzar un objetivo o meta en el contexto de una organización [Geo95]. Para cubrir estas necesidades es la tecnología Workflow quien ofrece un marco adecuado para abordar el problema.

Son varias las definiciones que se pueden encontrar sobre el término Workflow, la WFMC (Workflow Management Coalition) lo define como: “La automatización de un proceso de negocio, total o parcial, en la cual documentos, información o tareas, son pasadas de un participante a otro, de acuerdo a un conjunto de reglas establecidas” [Workflow1011].

Independiente de esta definición existen tres conceptos claves dentro de lo que concierne a los Workflows. Estos conceptos son: *Actividad, coordinación y personas*. “Un Workflow crea decisiones basadas en las condiciones del flujo determinadas durante el diseño, notifica a las personas involucradas en el proceso, indica el estado en que se encuentran una actividad, establece el trabajo que estas personas deben desarrollar y puntualizar” [Workflow1011].

Al igual que la evolución de la informática, en general, la evolución de Workflow está ligada con el cambio en los objetivos centrales de cada época. Si se resume la evolución de la informática en las últimas décadas, se puede ver como han cambiado los objetivos a seguir. En la década de los 60's y 70's el gran objetivo era resolver grandes cantidades de cálculo financiero y estadístico, de manera eficiente. En los 80's se buscaba mejorar el manejo y administración de las bases



de datos y en los 90's surge la necesidad de entender y poder manejar eficientemente el Workflow con el propósito de sacarle el mayor provecho. Si se mira la actuación de Workflow dentro de estas etapas, se puede identificar lo que sería un Workflow Manual en la década de los 60's y 70's, el Workflow Automatizado en la década de los 80's y lo que ofrece el Workflow en la actualidad.

En la década de los 60's y 70's se puede ver que antes que la informática se integrara al trabajo cotidiano, éste era realizado manualmente combinando toda la información en distintas carpetas físicas. En este ambiente era bastante difícil determinar el estado de una carpeta, de un documento físico y del proceso a seguir. Se manejaban grandes cantidades de documentos en forma manual, con los errores humanos que esto conlleva. Por esto se puede identificar un Flujo de trabajo manual o *Workflow Manual* de esa época. Luego, surge la necesidad de reemplazar las actividades manuales por actividades automáticas, es decir, se busca tener un mayor control y coordinación sobre toda la información que se maneja para llevar a cabo las tareas en las organizaciones.

Cuando llega la década de los 80's se puede apreciar la existencia de diversos sistemas de información, donde se maneja y administra toda la información necesaria para llevar a cabo la producción de las empresas u organizaciones. Se logran automatizar ciertas tareas que antes se realizaban manualmente. Es por esto, que se puede hablar de *Workflow Automatizado*. A finales de esta década se buscaba mejorar el flujo de información, el desafío que se plantea es obtener la información rápida y eficientemente. Es allí donde surgen las necesidades de incrementar la eficiencia, optimizar la productividad, acortar los tiempos de los procesos, tener un control sobre estos, así como también reducir los costos y mejorar la gestión. Todo esto surge como consecuencia del incremento de la

competitividad y de la exigencia de mejores productos, dentro de un mercado que avanza a gran velocidad.

El progreso tecnológico del Siglo XX y en especial, los ocurridos en los últimos años, han producido un gran impacto en la forma en que las organizaciones realizan su trabajo. Los avances de la computación y las telecomunicaciones han contribuido enormemente a mejorar las tareas que se desarrollan en las organizaciones, facilitando el ambiente cooperativo y aumentando la productividad de los grupos de trabajo como se muestra en el modelo de Workflow de Action Technologies<sup>1</sup>

Como respuesta a este cambio, surge en el mercado, un conjunto de productos de software orientado al desarrollo del trabajo grupal como lo son las aplicaciones o productos con tecnología “Workflow”. Estos productos han cambiado el enfoque de la tecnología de la información hacia el enfoque de la tecnología Workflow. Los sistemas que dan soporte a la definición de un flujo de trabajo y a su posterior ejecución, se denominan Sistemas de Gestión de Flujos de Trabajo (SGFT).

Los SGFT surgieron primero en la industria y más tarde se convirtieron en área de investigación (como en el caso de las Bases de Datos Orientadas a Objetos [DeW91]). Esto ha motivado la aparición en el mercado de numerosos productos, catalogados como SGFT, que resuelven problemas concretos pero, en los que en muchas ocasiones, hace falta una base científica que permita resolver ciertas limitaciones existentes. Entre ellas podemos citar problemas de escalabilidad, interoperabilidad, robustez, falta de soporte metodológico al modelado y soporte al análisis de registros de ejecución. Más adelante se identificarán las limitaciones de los SGFT que ofrece el mercado en la actualidad.

---

<sup>1</sup> Action technologies & Swiss Re Win Workflow medel.  
Rich Berman 2005. Disponible en: <http://www.actiontech.com/bpm/ABIM.htm>

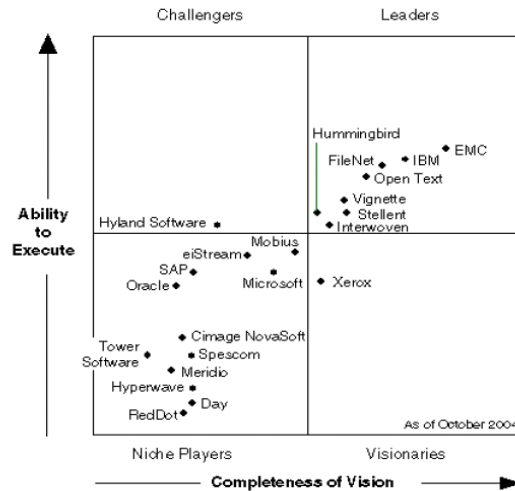
Respecto a los productos existentes en el mercado, no todos entran dentro de la misma categoría, ni todos ellos proporcionan las mismas prestaciones a la hora de modelar o ejecutar un flujo de trabajo. Existen diversos tipos de SGFT dependiendo de las características del flujo de trabajo que gestionen. Algunos de estos productos son: MQSeries Workflow de IBM, InConcert, Staffware, Lotus Notes, Bizflow 2000, Eastman Enterprise Workflow, Dolphin, Filenet con Panagon Workflow y SERfloware. Posteriormente se detallarán las características básicas de los SGFT.

El estudio sectorial, basado en la herramienta analítica conocida como “Cuadrantes mágicos” de Gartner<sup>2</sup>, describe los puntos fuertes y débiles de los fabricantes de SGFT en mercados específicos, situando a las compañías en los cuadrantes según su visión del mercado y su capacidad de ejecución. Para Garther, las empresas visionarias son aquellas que presentan un enfoque claro sobre la dirección del mercado y que orientan sus esfuerzos en este sentido, y que todavía están en condiciones de optimizar sus servicios. Según Gartner los líderes tienen los mejores valores combinando la visión de mercado y su capacidad de ejecución. En la figura 1 en el cuadrante superior izquierdo Gartner ubica los fabricantes de herramientas que satisfacen las necesidades de las empresas de hoy, y tienen la capacidad de anticiparse a futuras demandas. Para Gartner las empresas líderes son aquellas que están obteniendo unos buenos resultados en la actualidad, tienen una visión clara del mercado y hacia dónde se dirige, y día a día ponen los medios necesarios para mantener su posición de liderazgo mediante nuevas herramientas y funcionalidades.

---

<sup>2</sup> Gartner Inc. Organización consultora dedicada a proporcionar investigación objetiva, defendible y creíble para ayudar a los inversionistas de la tecnología a tomar decisiones de inversión, por medio de los resultados arrojados de las diferentes investigaciones realizadas [GAR05]

**Figura 1. Cuadrante mágico de Gartner**



Fuente: [GAR05]

## 1.2 EVOLUCIÓN DE WORKFLOW

Durante algunos años, muchos productos en el mercado de Tecnologías de Información han brindado soporte sobre algunos aspectos funcionales de Workflow, sin embargo, sólo hasta hace relativamente poco tiempo se ha empezado a reconocer la importancia de esta tecnología en sí misma. La evolución de Workflow como tecnología ha integrado distintas áreas, como se describen en [Khoshafian], algunas de estas áreas son las siguientes

**Procesamiento de Imágenes:** Los Workflows se han asociado de manera cercana a sistemas de imágenes y muchos de ellos tienen capacidad de flujos de trabajo, ya sea integrada al sistema o suministrada dentro de un producto Workflow específico. Muchos procedimientos de negocio implican la interacción con información en papel, la cual puede ser necesaria capturar como imagen digitalizada, como parte de un proceso de automatización. Una vez que la información en papel ha sido capturada de manera electrónica como datos de

imagen, suele ser necesario que estos pasen entre una serie de participantes, por distintas razones, lo cual podría implicar la interacción con otras aplicaciones de Tecnología de información, creando de este modo, un requisito para la funcionalidad de Workflow.

**Gestión de Documentos:** La tecnología de gestión de documentos está relacionada con la administración del ciclo de vida de los documentos electrónicos. Cada vez más, esta tecnología incluye herramientas para gestionar repositorios de documentos distribuidos dentro de una organización como un recurso compartido con herramientas para enrutar documentos a diferentes personas o incluso separar partes de estos para tener acceso a la información o hacer actualizaciones de acuerdo con sus roles específicos. El documento puede hacer parte de un proceso de negocio determinado, en donde parte del personal que lleva a cabo actividades separadas requiere acceso al documento, de acuerdo, con una secuencia particular y con algunas reglas procedimentales.

**Correo Electrónico y Directorios:** El correo electrónico ofrece herramientas poderosas para distribuir información entre los individuos de una organización o entre organizaciones. El uso de mecanismos de directorios no sólo proporciona una manera de identificar los participantes individuales de un dominio de correo, sino también puede ser una manera de registrar información sobre los atributos del usuario individual, tales como el rol dentro de la organización u otros atributos relacionados con procedimientos de negocio. Así, los sistemas de correo electrónico han estado evolucionando hacia la funcionalidad de flujos de trabajo a través de la adición de comandos de enrutamiento en respuesta a alguna forma de procedimiento de negocios identificada.

**Aplicaciones Colaborativas (Groupware):** La industria de aplicaciones colaborativas ha introducido un amplio rango de aplicaciones de software

diseñadas para apoyar y mejorar las interacciones entre grupos o individuos. Inicialmente, muchas de estas aplicaciones apoyaban mejoras en grupos que trabajaban mediante procesos informales, introduciendo de manera temporal, las aplicaciones de programación de actividades. Ya que el alcance de estas aplicaciones se ha extendido a interacciones grupales de negocio más formales y enfocadas, ha habido una necesidad cada vez mayor de ofrecer un marco de procedimiento más formal y controlable, para apoyar el uso de aplicaciones colaborativas. La tecnología Workflow ofrece una solución a este tipo de requerimiento.

Aplicaciones basadas en Transacciones: Durante muchos años se han desarrollado aplicaciones para apoyar cierto tipo de procedimientos de negocio (“Transacciones”) utilizando herramientas de gestión de transacciones y/o software de gestión de bases de datos. Aunque el estilo de trabajo inicial era centralizado, este software ha ido permitiendo cada vez más la distribución de aplicaciones basadas en transacciones a lo largo de un número de plataformas informáticas. Las aplicaciones basadas en transacciones generalmente muestran importantes características de robustez y soporte a propiedades “atómicas” de la transacción; sin embargo, generalmente no muestran una separación entre la lógica del proceso de negocio y la invocación de distintas aplicaciones que pueden ser necesarias para apoyar actividades individuales dentro de un proceso. Con el tiempo, esto ha ido creando la necesidad de consolidar las capacidades de los Workflows para controlar los procedimientos de negocio con la posibilidad de invocar aplicaciones tradicionales e integrarlas con las partes adecuadas de un proceso de negocio.

Software de Soporte para la Gestión de Proyectos: El software para manejar el desarrollo de aplicaciones basadas en tecnologías de información, ha proporcionado una forma de funcionalidad de Workflow dentro del entorno del

proyecto, para “transferir” tareas entre individuos y definir rutas de información dando soporte a estas tareas. En algunos casos este tipo de software ha sido generalizado para dar soporte a una visión de procesos más amplia y orientada al negocio y a un rango más amplio de aplicaciones, ofreciendo una capacidad de Workflow o de flujo de trabajo más general.

BPR (Reingeniería de Procesos de Negocio) y Herramientas Estructuradas de Diseño del Sistema: Las herramientas de reingeniería de procesos de negocio han ofrecido un soporte basado en tecnologías de información (TI) a las principales actividades de análisis, modelamiento y redefinición de los procesos de negocio de una organización, a los efectos potenciales de un cambio en dichos procesos o los roles y responsabilidades organizacionales. Esto puede incluir el análisis de las estructuras del proceso y los flujos de información que lo soportan, los roles de las personas o unidades organizacionales dentro del proceso y las acciones que se realizan en respuesta a diferentes eventos. El objetivo de estas herramientas es facilitar la implementación del proceso con una infraestructura de TI de soporte, para controlar los flujos de trabajo y las actividades asociadas en el proceso de negocio.

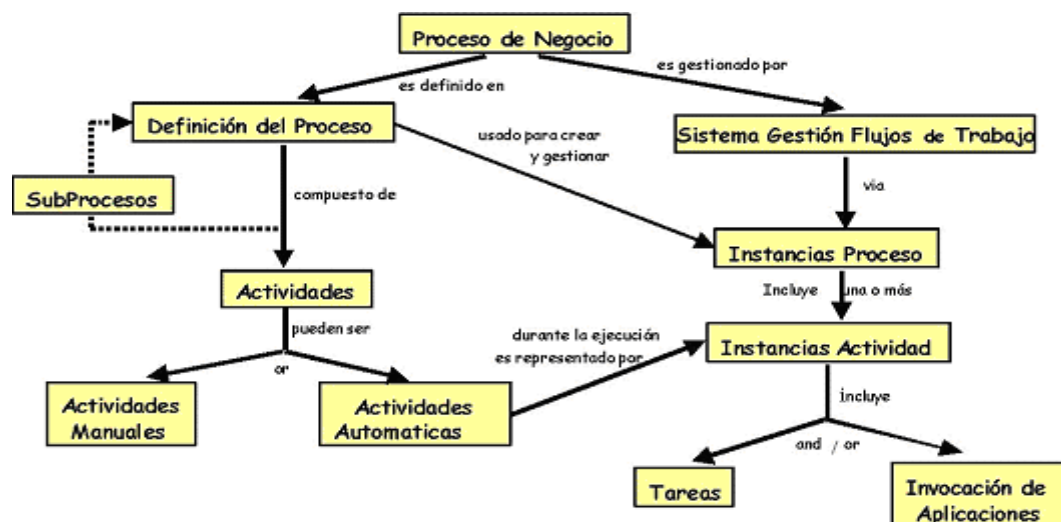
Separación de la funcionalidad del flujo de trabajo: El mercado Workflow ha evolucionado y seguramente lo seguirá haciendo. Existe un amplio rango de productos centrados en uno o más aspectos particulares sobre el flujo de trabajo. Algunos se ofrecen en conjunto con otras áreas de tecnología, como el procesamiento de imágenes o la gestión de documentos; otros pueden servir para un propósito más general. Esta multiplicidad de productos permitirá un mayor número de opciones en circunstancias particulares de implementación. Sin embargo, también aumenta la necesidad de crear estándares dentro de la industria, con el fin de permitir que diferentes productos funcionen juntos y se integren dentro de una arquitectura global consistente.

## CAPITULO 2. SISTEMAS DE GESTIÓN DE FLUJOS DE TRABAJO

Un SGFT es “un sistema que define, crea y gestiona la ejecución de flujos de trabajo mediante el uso de software, siendo capaz de interpretar la definición del proceso, interactuar con los participantes y, siempre que se requiera, invocar el uso de herramientas y aplicaciones” [WFM99a].

De acuerdo con la definición anterior, en un SGFT existen dos actividades claramente diferenciadas, aunque con relaciones entre ellas. Por una parte está la definición del flujo de trabajo que implementa al proceso de negocio, en lo que llamaremos modelado del flujo de trabajo, y por otra parte está la animación o ejecución de dicho modelo, también conocido en la literatura con el término inglés *enactment*. La figura 2 muestra los conceptos básicos y la terminología asociada tanto para la fase de modelado como para la fase de ejecución.

**Figura 2. Conceptos básicos y terminología Workflow**



Fuente: [WFMC1003]



La especificación del flujo de trabajo consiste en definirlo como un conjunto de actividades relacionadas, más un conjunto de criterios que indican el inicio y fin de un proceso y de sus componentes, e información adicional sobre cada actividad, como lo son: participantes, invocación de aplicaciones, datos, etc.

Las actividades describen trozos de trabajo y constituyen pasos o tareas dentro del proceso. Éstas pueden ser manuales o automáticas, en el sentido de que puedan requerir o no la participación humana, así como invocar aplicaciones externas que realicen parte o todo el trabajo asignado a dicha actividad.

La ejecución del flujo de trabajo en un SGFT consiste en la creación, manipulación y destrucción de instancias del proceso, que representen al flujo de trabajo de acuerdo con la especificación previa. Cada instancia del proceso tendrá una identidad visible externamente y un estado interno que representa su progreso hacia la finalización y su estado con respecto a las actividades que lo componen. Cada vez que la ejecución del proceso suponga la invocación de una actividad, según la definición del mismo, se crea una instancia actividad que la representa. Dicha instancia se encarga de ejecutar las acciones asignadas, accediendo a los datos que sean necesarios e invocando la aplicación externa correspondiente, si así lo requiere la actividad.

Además del *proceso* o flujo de trabajo como componente más importante, existen otros dos aspectos que se pueden considerar. En primer lugar, cuando algunas actividades pueden requerir participación humana, se puede incluir no sólo la información de quién realiza dicha tarea, sino también modelar toda la estructura organizativa del personal. En segundo lugar, y de forma similar a lo anterior, la utilización de datos por parte de las actividades y la posibilidad de invocar aplicaciones externas obliga, en cierto modo, a conocer y modelar la propia infraestructura de tecnologías de la información existentes en la organización.

## 2.1 CLASIFICACIÓN DE LOS SISTEMAS DE GESTIÓN DE FLUJOS DE TRABAJO

Los elementos que se deben considerar en la especificación de un flujo de trabajo son muchos y muy variados, como ya se ha comentado anteriormente. Por ello, no existe una única clasificación de los SGFT, sino que existen varias, dependiendo del criterio a utilizar. A continuación se encuentran algunas de las clasificaciones:

**a. Según la complejidad del grafo que representa al flujo de trabajo [Geo95]:** Existen dos tipos de flujo de trabajo para la coordinación entre las diversas actividades:

***Ligeramente estructurado.*** En este caso, el grafo que representa la coordinación entre las actividades es prácticamente lineal, y cada una de ellas se van ejecutando una a continuación de otra.

***Altamente estructurado.*** El grafo que representa la coordinación entre las actividades ya no es lineal, sino que se representan ejecuciones en paralelo de actividades, sincronización de actividades, etc.

**b. Según el grado de participación humana en el flujo de trabajo [Geo95]:** Se pueden clasificar en:

***Orientados a personas.*** Cuando la participación humana en la ejecución del SGFT es importante. Estas personas serán agentes que inician o realizan las actividades. Por su naturaleza, la mayor parte de las actividades que conforman el flujo de trabajo son manuales.

***Orientados a sistemas.*** La participación humana suele ser menor y el proceso está altamente automatizado, por lo que la mayor parte de las actividades del flujo

de trabajo son actividades automáticas.

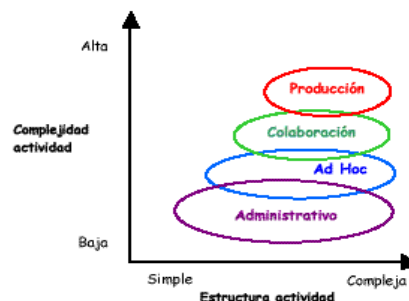
**c. De acuerdo con la tecnología en la que se basa el SGFT:** Se clasifican en: ***Centrados en el correo electrónico.*** En estos SGFT se utiliza el *email* como medio de comunicación.

***Centrados en el documento.*** La principal característica es que los documentos circulan e interaccionan con aplicaciones externas. Predominan los aspectos de gestión de documentos.

***Centrados en el proceso.*** En este caso, lo importante es el propio proceso. Los mecanismos de comunicación entre las actividades y/o los datos se implementan por encima de la base de datos, proporcionándose interfaces de interacción.

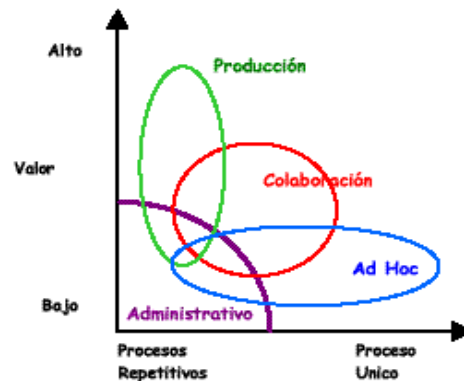
**d. Clasificación tipos de Workflow:** Una clasificación de SGFT es la que explica por qué el mercado de Workflow ha sido dividido en varios segmentos según el valor del proceso a manejar y si este proceso es repetitivo o no. La figura 3 y la figura 4 muestran cómo se sitúa cada tipo de SGFT de acuerdo a los criterios establecidos.

**Figura 3. Clasificación de los SGFT según complejidad y estructura de las actividades involucradas.**



Fuente: [Geo95]

**Figura 4. Clasificación de los SGFT según la similitud de los procesos de negocio y su valor en la Organización.**



Fuente: [Geo95]

**Administrativos.** Este tipo de SGFT es el que modela los procesos burocráticos de una organización. La función básica asociada al mismo, es el *procesamiento de formularios*. El SGFT se encarga de activar la ejecución de las actividades (la mayor parte de las cuales son manuales e interviene un agente en ellas), recoger las respuestas (datos) y obtener el formulario (que normalmente es una o varias actividades automáticas, que recopilan y procesan la información obtenida). Por ejemplo, la matrícula universitaria o la obtención de certificados.

La principal característica de las actividades involucradas en el proceso es que se trata de actividades repetitivas y de baja complejidad. La estructura del grafo que representa el proceso puede variar de baja a alta, aunque en muchas ocasiones el grafo de tareas es lineal.

**Ad Hoc.** Estos SGFT son muy similares a los administrativos, pero con la

característica de ser utilizados para tratar situaciones únicas o excepcionales, en lugar de procesos burocráticos perfectamente establecidos. El SGFT informa sobre el estado de ejecución de cada actividad y la participación humana es esencial.

Las actividades involucradas en el proceso suelen ser únicas, en el sentido de que muchas de ellas sólo se realizan una vez y su complejidad oscila de baja a media. La principal dificultad en este tipo de SGFT está en construir el grafo de coordinación y cooperación entre actividades.

Un ejemplo de este tipo es el proceso para la aceptación de artículos en una revista desde el punto de vista del autor. Cada revista tiene su propio proceso a seguir, y en este caso, seguir la pista por parte del autor de cada proceso de aceptación por separado no es complicado, pero sí en caso de estar todos juntos y ejecutándose simultáneamente (envío simultáneo a varias revistas).

***Producción.*** Modelan e implementan los procesos de negocio críticos de la organización, es decir, los procesos directamente relacionados con la función principal de la misma.

Los SGFT de este tipo son sistemas complejos que se ejecutan sobre entornos heterogéneos, y en los que suelen participar como agentes gran variedad de personas y Organizaciones. Normalmente se requiere la ejecución de transacciones para el acceso a la información, que pueden encontrarse en diferentes sistemas. Finalmente, cabe destacar que la monitorización del estado en el que se encuentra cada actividad es muy importante en este tipo de sistemas, así como el posterior análisis estadístico de toda esta información, que va a permitir, entre otras cosas, mejorar el proceso.

Las actividades involucradas en el proceso son repetitivas (cientos o miles de instancias en ejecución), pero de alta complejidad. La estructura del grafo que representa el proceso suele ser más bien compleja.

Dentro de los ejemplos de esta clasificación se encuentran la integración de software preexistente (*legacy applications*), los préstamos bancarios, préstamos y devoluciones en general, expedición de pólizas de seguros, etc.

**Colaboración.** Este tipo de SGFT se caracteriza principalmente por la participación de distintas personas y las interacciones que tienen lugar entre ellas, de tal forma que la mayor parte de la coordinación en realidad la realiza el hombre. Otra característica importante es la existencia de ciertas actividades sobre las que se pueden realizar varias iteraciones hasta que se alcance un cierto consenso por parte de todos los participantes; una vez alcanzado, la ejecución continúa siempre hacia adelante. Son sistemas muy dinámicos que en muchas ocasiones se definen conforme se avanza en el proceso.

Las características enunciadas anteriormente hacen que no exista un consenso total sobre si realmente se trata o no de SGFT, especialmente cuando las características anteriores se llevan al límite, ya que la mayor parte de la coordinación la realiza el hombre y el sistema se limita a proporcionar una buena interfaz para las interacciones (normalmente vía correo electrónico).

## **2.2 LIMITACIONES DE LOS SISTEMAS DE GESTIÓN DE FLUJOS DE TRABAJO**

Existen en el mercado numerosos productos comerciales catalogados como SGFT. No todos ellos reúnen las mismas características, sino que muchas veces se centran en proporcionar funcionalidades específicas que encajan en alguna de las clasificaciones anteriormente vistas. Además cada uno de estos productos han

evolucionado incorporando nuevas tecnologías y convirtiéndose muchas veces en productos para aplicaciones muy específicas.

Las principales limitaciones de los SGFT se pueden resumir en las siguientes [Alo97]:

- ⌚ Los SGFT existentes no son totalmente compatibles, haciéndose prácticamente imposible unir diferentes sistemas que proporcionan diferentes funcionalidades en uno solo. Las incompatibilidades existentes no sólo tienen que ver con la sintaxis o la plataforma en la que se ejecutan, sino que muchas veces dichas incompatibilidades se refieren al propio modelo de ejecución del flujo de trabajo. Esta situación hacen que los diferentes productos existentes se consideren como “islas” de la automatización de procesos.
- ⌚ En muchas ocasiones, estos productos fueron diseñados para pequeños grupos de usuarios. Cuando estos productos se han necesitado utilizar a gran escala, han aparecido numerosas restricciones derivadas del propio diseño del producto, tales como: pobre soporte a la comunicación, una única base de datos donde se almacena toda la información, etc. Estas restricciones obligan en numerosas ocasiones a rediseñar por completo el SGFT.
- ⌚ La falta de control transaccional en muchos productos también es una limitación importante. Ante un fallo, no siempre aseguran una recuperación correcta del sistema, sino que muchas veces queda en manos de los propios mecanismos de recuperación de la base de datos. Cuando se presenta un fallo no se puede hacer rollback por que se pierde el control entre las partes

- ⌚ No existe hasta el momento una formalización de la noción de flujo de trabajo, sino que en la mayoría de las ocasiones simplemente se proporcionan descripciones informales de cuáles son sus componentes.
- ⌚ No existe soporte metodológico para los procesos de modelado de flujos de trabajo, ni tampoco para el análisis de los mismos [She96].

Casi todas las limitaciones mencionadas anteriormente se deben a la falta de un Modelo de Referencia común para todos los SGFT; a continuación se describe el Modelo de Referencia propuesto por la *Workflow Management Coalition (WfMC)*

## **2.3 ESTANDARIZACION DE LOS SGFT**

La Workflow Management Coalition (WFMC) es una organización internacional, sin ánimo de lucro, que se establece en agosto de 1993 y reúne a un grupo de vendedores, usuarios, consultores, analistas y grupos de investigación. Su principal objetivo es promover el uso de SGFT mediante el establecimiento de estándares que faciliten la creación, desarrollo y análisis de estos sistemas.

### **2.3.1 Necesidad de estandarizar**

Se estima que actualmente los distintos productos Workflow como SGFT que existen en el mercado sobrepasan los cien. Cada uno de ellos se enfoca sobre distintos aspectos funcionales, como herramientas de diseño visual, en las cuales se ofrecen ciertos diagramas para representar la realidad. Otros se enfocan en la integración de los datos con las aplicaciones.

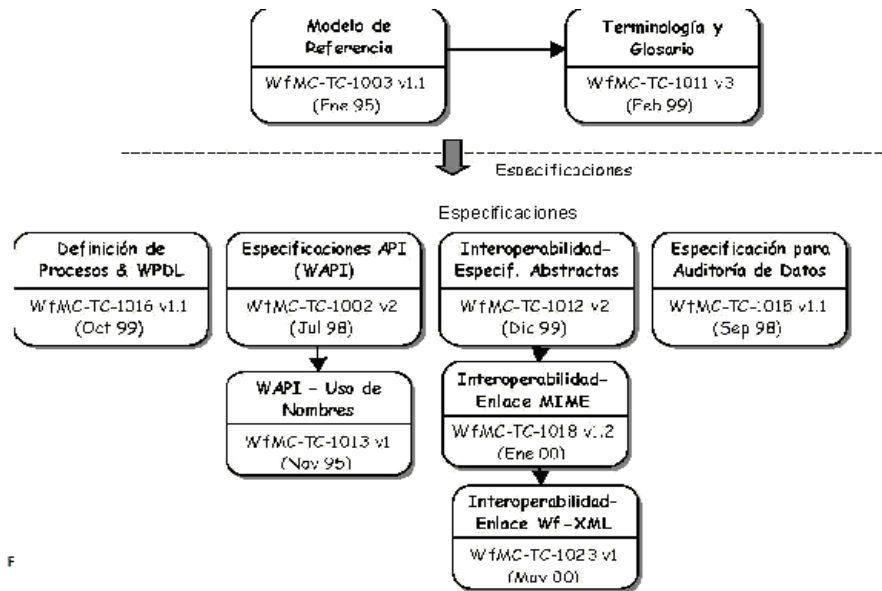


Las organizaciones constantemente rediseñan sus procesos, esta metodología es denominada Reingeniería de los Procesos de Empresas. Esto puede ser causa de cambios organizacionales, legislativos, cambios en los objetivos del negocio, entre otros. En esta situación, muchas veces es necesario relacionarse con otras organizaciones. Pero para poder hacer esto debe existir la posibilidad de que los productos de un vendedor puedan comunicarse con los de otro, pues ya que es claro que cada empresa u organización comprará los productos que crea convenientes para su caso.

Por lo anterior surge la necesidad de estandarizar la forma de comunicación entre los distintos componentes de un producto Workflow con el fin de poder tener flexibilidad a la hora de operar con distintos productos. Esta necesidad se justifica por las proyecciones que se tienen actualmente sobre la penetración de la tecnología Workflow en el mercado en los próximos años.

La WfMC ha definido una serie de estándares (ver figura 5). De ellos, se destaca el *Modelo de Referencia* [Hol95] que describe la arquitectura básica de un SGFT y que en conjunto con el de *Terminología y Glosario* [WFM99a] constituyen el material básico. Con base en ellos se han definido otras especificaciones con una finalidad mucho más concreta, como son: interoperabilidad entre SGFT [WFM99c], modelo para la definición de procesos [WFM99b], invocación a través de API (*Application Programming Interface*) [WFM98a] y análisis de datos [WFM98b].

**Figura 5. Estándares de la WfMC**



**Fuente:** [WfMC1003].

### 2.3.2 Modelo de referencia de la WfMC

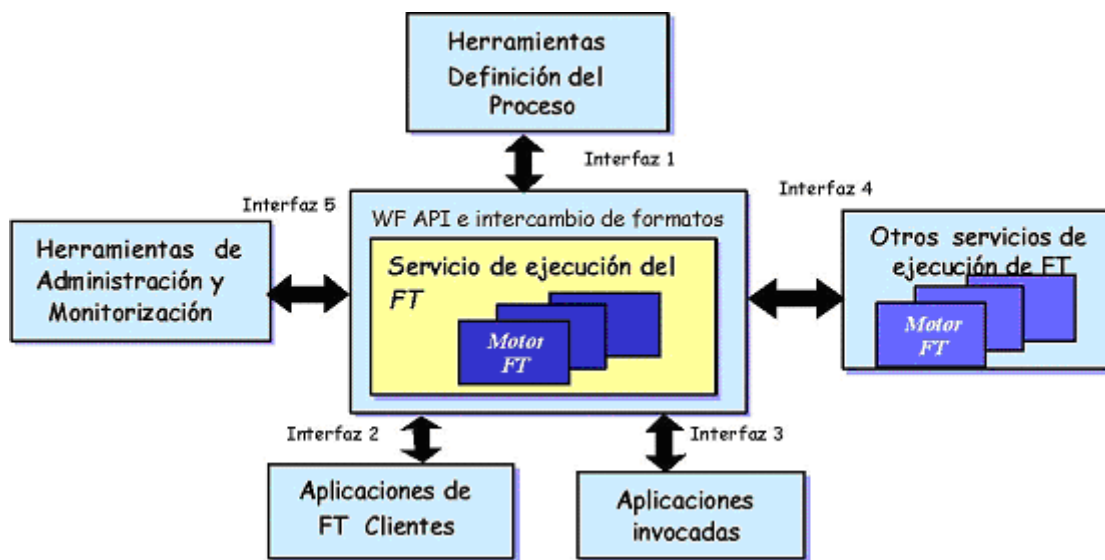
El modelo de referencia de Workflow fue desarrollado desde estructuras genéricas de aplicaciones Workflow, identificando las interfaces con estas estructuras de forma que permita a los productos comunicarse con distintos niveles. Todos los sistemas Workflow contienen componentes genéricos que interactúan de forma definida. Para poder tener cierto nivel de interoperabilidad ente los diversos productos Workflow, es necesario definir un conjunto de interfaces y formatos para el intercambio de datos entre dichas componentes [WfMC1003].

En primer lugar se han identificado las distintas áreas funcionales y posteriormente se han desarrollado especificaciones para la implementación de las mismas,

asegurándose la interoperabilidad entre distintos SGFT y su integración con otras aplicaciones informáticas.

Este modelo ha sido desarrollado a partir de la *arquitectura de una aplicación de sistema de flujo de trabajo genérica*, identificando sus componentes y las diferentes interfaces que permiten comunicación a diferentes niveles (ver figura 6).

**Figura 6. Modelo de Referencia de un SGFT**



Fuente: [WFM1003].

El componente central es el **servicio de ejecución del flujo de trabajo** que se encarga de crear, gestionar y ejecutar cada una de las instancias del modelo de flujo de trabajo. En este componente es donde se encuentra el motor del SGFT, que proporciona la ejecución, propiamente dicha, de cada instancia. En caso de estar en un entorno de ejecución de flujo de trabajo distribuido, pueden existir diferentes motores de flujo de trabajo que controlen distintas partes de la ejecución del proceso. La comunicación de este componente con el resto se realiza a través

de lo que la WfMC denomina *WAPI (Workflow APIs)*, es decir, la interfaz para la programación de aplicaciones de flujo de trabajo [WFM98a].

Otro de los componentes son ***las herramientas de definición del proceso*** o flujo de trabajo, las cuales permiten modelar, describir y documentar un determinado flujo de trabajo, proceso de negocio o proceso en general. Estas herramientas pueden ser totalmente informales (lenguaje natural, lápiz y papel) o mucho más sofisticadas y formalizadas (interfaces gráficas con un modelo subyacente bien definido). Se debe especificar la lógica del proceso, las actividades que lo componen, los participantes humanos, aplicaciones invocadas, datos utilizados, etc. La ***interfaz 1*** permite la comunicación entre este componente y el servicio de ejecución del flujo de trabajo.

Además del propio servicio de ejecución del flujo de trabajo y las herramientas para la definición del proceso, tenemos otro componente denominado ***aplicaciones clientes del flujo de trabajo***, que representa las entidades software utilizadas por el usuario final en aquellas actividades que requieren participación humana para su realización. Si este componente se separa de lo que es el propio componente de ejecución, es necesaria una interfaz (***interfaz 2***) que defina y maneje claramente el concepto de lista de trabajos o *worklist*, como una cola de trabajo asignado a un usuario o a un grupo de usuarios por el propio motor de ejecución del flujo de trabajo.

El componente ***aplicaciones invocadas*** representa software o aplicaciones ya existentes que un SGFT puede utilizar para la realización de ciertas actividades, teniendo en cuenta que, en principio, dichas aplicaciones se pueden encontrar en cualquier plataforma o lugar de la red. La ***interfaz 3*** permite la comunicación entre este componente y el servicio de ejecución del flujo de trabajo, no sólo a nivel de

invocación del mismo, sino de transformación de datos en formatos entendibles por ambos componentes. Una posible solución se obtiene a través de lo que se denomina *agente aplicación*, de modo que el servicio de ejecución del flujo de trabajo se comunica con las funciones estándar de dicho agente aplicación y éste define interfaces específicas para cada tipo de aplicación invocada.

La interoperabilidad entre SGFT está representada por el componente denominado ***otros servicios de ejecución de flujo de trabajo***, siendo la **interfaz 4** la que permite dicha comunicación. En este caso, la WfMC ha desarrollado un conjunto de escenarios de interoperabilidad que van desde la conexión a nivel de actividades simples hasta todo un completo intercambio de definición de procesos y datos [WFM99c].

Finalmente, el componente ***herramientas de administración y monitorización*** permite que distintos servicios de ejecución de flujo de trabajo compartan las mismas funciones de administración y monitorización del sistema, como pueden ser, por ejemplo, la gestión de usuarios, el control de los recursos y la supervisión del estado de todo el proceso.

## **PARTE II PRELIMINARES PARA EL DESARROLLO DE UNA METODOLOGÍA DE AUTOMATIZACIÓN DE PROCESOS WORKFLOW**

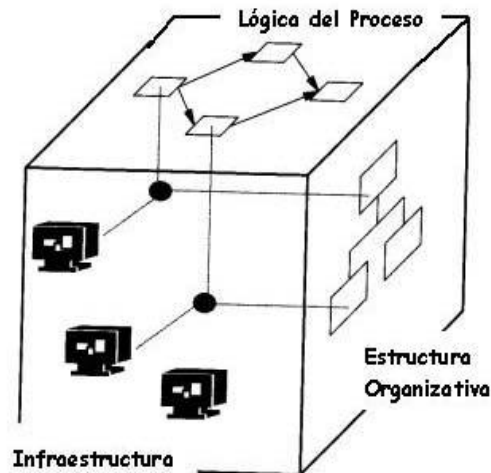
### **CAPITULO 3. ELEMENTOS QUE COMPONEN UN FLUJO DE TRABAJO**

Una metodología para el desarrollo de procesos Workflow debe expresar qué es un flujo de trabajo y captar todos los aspectos o componentes del mismo. En este capítulo se presentan los componentes básicos y esenciales de un flujo de trabajo los cuales están de acuerdo con las especificaciones de la WfMC y la de otros trabajos relevantes en este campo, tales como [Cur92], [Cas95], [Alo98] y [Sad99]. Es importante destacar que, aunque se recogerán los aspectos más importantes de un flujo de trabajo, pueden existir ciertos dominios de aplicación en los cuales puedan aparecer nuevos elementos que hagan necesario adicionarlos o enriquecer este trabajo con nuevas propiedades.

#### **3.1 DIMENSIONES DE UN FLUJO DE TRABAJO**

Se considera que un flujo de trabajo tiene tres dimensiones básicas, que son ortogonales entre sí [Ley00]. Estas son: *Lógica del proceso*, *Estructura organizacional* e *Infraestructura de tecnologías de la información*. La figura 7 ilustra estas tres dimensiones.

**Figura 7. Dimensiones de un flujo de trabajo**



Fuente: [Ley00]

*La lógica del proceso* describe qué tareas se realizan y en qué orden. Las tareas pueden ser actividades concretas o subprocesos, que se pueden invocar de forma local o remota, dentro de la misma organización o en otra distinta. En cuanto al flujo de control, éste puede ser secuencial o paralelo, pudiéndose establecer condiciones que determinen el camino a seguir dentro del proceso.

*La estructura organizativa* se refiere a unidades organizacionales o departamentos, relaciones entre ellas, usuarios que forman parte de las mismas y roles que éstos pueden desempeñar. Esta dimensión da cuenta de quién realiza cada tarea.

*La infraestructura de tecnologías de la información* disponible en la organización, tal como programas o aplicaciones que se utilizan para realizar una determinada tarea, representa los recursos asociados al flujo de trabajo.

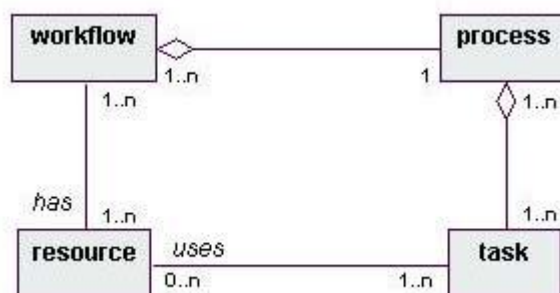
En un flujo de trabajo confluyen tres dimensiones, que dan cuenta de **qué se ejecuta, quién lo ejecuta y con qué se ejecuta**.

### 3.2 COMPONENTES DE UN FLUJO DE TRABAJO

Un *flujo de trabajo* está formado por un *proceso* y tiene además asociado un conjunto de *recursos*. Lógicamente existe una relación entre el proceso que constituye el flujo de trabajo y los recursos asociados al mismo. El proceso está formado por un conjunto de *tareas* y son éstas las que realmente utilizan los recursos.

La figura 8 muestra una visión global de los principales componentes de un flujo de trabajo y sus relaciones, en notación UML. Se puede observar que el flujo de trabajo es modelado como una clase (denominada *Workflow*). De la misma forma, sus componentes también son clases. Concretamente, la clase *process* representa el proceso, la clase *resource* representa los recursos y la clase *task* representa las tareas del proceso. La utilización de los recursos por parte de las tareas que forman el proceso, queda representado mediante la asociación entre la clase *resource* y la clase *task*.

**Figura 8. Componentes de un Flujo de Trabajo**



Fuente: [WFMC1003].



A continuación se detalla cada uno de estos componentes.

### 3.2.1 Proceso

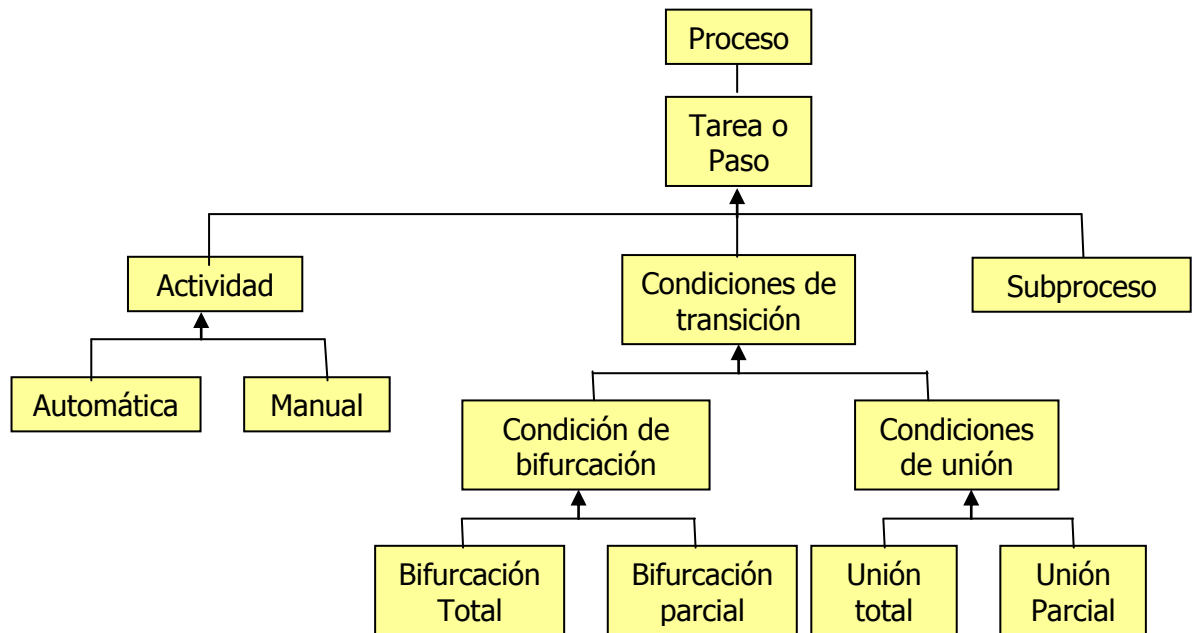
Un proceso está formado por un conjunto de *tareas* conectadas mediante *flujos de control*. Este flujo de control establece el orden correcto de ejecución de las distintas tareas que componen el proceso. Las tareas pueden ser de tres tipos: *actividades*, *subprocesos* y *condiciones de transición*. Las *actividades* y *subprocesos* especifican cada uno de los pasos que componen el proceso a automatizar. Las *condiciones de transición* permiten establecer puntos de bifurcación y/o de unión en el flujo de control. Además de las tareas conectadas mediante flujos de control, todo proceso tiene asociada la siguiente información:

*Identificador – Nombre – Descripción – Condición de inicio – Condición de finalización - Estado*

La condición de inicio del proceso es una condición que sólo cuando se evalúa a cierto, permite que el proceso se inicie realmente, es decir, que el control pase a la primera tarea del mismo. La condición de finalización, de igual forma, determina el estado de finalización del proceso; de modo que si se evalúa a cierto, el proceso finaliza con éxito y si se evalúa a falso, el proceso finaliza sin éxito. Finalmente, todo proceso tiene información acerca del estado en que se encuentra. Se requiere saber cuándo un proceso ha sido creado, cuándo se cumple la condición de inicio y el proceso pasa a estar activo, o por el contrario, si ésta no se cumple y el proceso no se ejecuta, cuándo el proceso inicia realmente su ejecución y cuándo finaliza, teniendo en cuenta si la finalización del mismo se ha producido con éxito, sin éxito, o bien si ha sido abortado por el usuario.

En la figura 9 se puede observar la jerarquía de generalización correspondiente a las tareas.

**Figura 9. Jerarquía de Generalización**

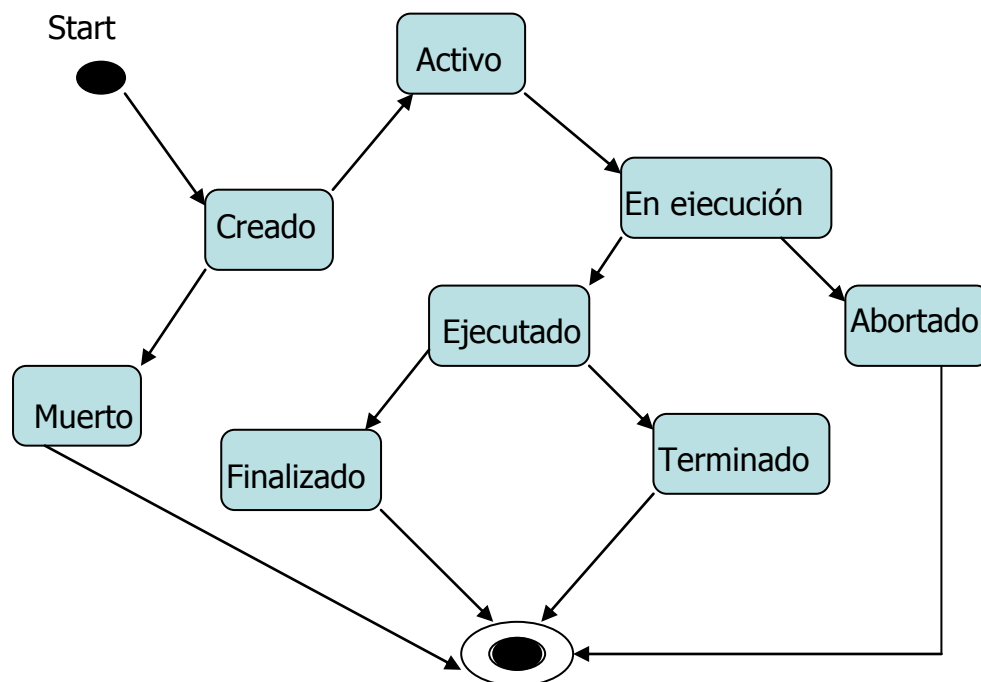


Los posibles estados en los que se puede encontrar un proceso, junto con las transiciones válidas entre ellos, permiten definir su diagrama de estados, como se muestra en la figura 10. Los posibles estados de un proceso son: *creado*, *activo*, *en ejecución*, *fin de ejecución o ejecutado*, *finalización con éxito o finalizado*, *finalización sin éxito o terminado*, *finalización por interrupción del usuario o abortado* y *muerto*. Como se puede apreciar en la figura 10, tras el evento de creación se alcanza el estado creado, en el cual se evalúa la condición de inicio y sólo en el caso que se evalúe a cierto pasará al estado activo. Por el contrario, si la condición de inicio no se cumple, el proceso no puede seguir con su ejecución y pasa al estado muerto. Cuando el proceso inicia realmente su ejecución, lo cual implica que se empiece a ejecutar la primera tarea que lo compone según el orden definido en el flujo de control, el proceso pasa a estar en ejecución. En este estado

pueden ocurrir dos cosas: que la ejecución del proceso finalice, con lo cual se alcanza el estado fin de ejecución, o bien que el usuario decida interrumpir la ejecución del proceso, y se alcanza el estado de finalización por interrupción del usuario. En el primer caso, se evalúa la condición de finalización, y el resultado del mismo determina el siguiente estado a alcanzar. Si la condición se evalúa a cierto, el estado alcanzado es de finalización con éxito; en caso contrario se alcanza el estado finalización sin éxito.

Como se puede apreciar, el cambio de estado no queda determinado únicamente por la ocurrencia de eventos, sino también por el cumplimiento o no de las condiciones de inicio y finalización. Éstas constituyen una parte importante de la definición y ejecución del proceso, determinando la mayor parte de los estados finales que dicho proceso puede alcanzar antes de que se destruya la instancia que lo representa.

**Figura 10. Diagrama de transición de estados**



Fuente: [WPMC1003].

### **3.2.1.1 Actividad**

Una actividad representa cada uno de los pasos o tareas básicas a realizar dentro de un proceso. Desde la perspectiva del proceso que se está modelando, toda actividad se considera atómica, es decir, que no se puede descomponer en otras actividades. En otro caso, se considera un subproceso, como se verá mas adelante.

Toda actividad tiene las siguientes propiedades:

*Identificador – Nombre – Descripción – Condición de inicio – Condición de finalización – Estado - Acciones*

La condición de inicio y la de finalización, tal como ocurre en el caso de un proceso, son condiciones que determinan cuándo una actividad realmente se inicia o finaliza, respectivamente. Al igual que con los procesos, se requiere saber cuándo una actividad está activa, está en ejecución, ha finalizado o ha sido interrumpida por el usuario. Por ello, los estados por los que puede pasar una actividad coinciden con los definidos para el proceso.

Existen dos tipos de actividades:

*Una actividad manual* es aquella que necesita al menos un participante humano para su realización; las actividades manuales tienen que ver con la toma de decisiones, completar información mediante formularios de datos, proporcionar información, etc. Algunos ejemplos de actividades manuales pueden ser, por ejemplo, autorizar el descuento de un producto a un cliente, revisar documentación enviada por un cliente.

*Una actividad automática* es aquella que normalmente no requiere participación humana para su realización, sino que es una determinada operación o conjunto de operaciones sobre un conjunto de datos, o bien, la invocación de una aplicación externa. Algunos ejemplos de actividades automáticas pueden ser, por ejemplo, lanzar una transacción sobre los datos o realizar cálculos estadísticos.

Por otra parte, en todo proceso se consideran dos actividades diferenciadas que marcan el inicio y final del mismo y son denominamos *actividad inicial* y *actividad final*, respectivamente.

En la figura 9 se puede apreciar el modelo conceptual de una actividad. En este modelo no se expresan las relaciones existentes entre las actividades y los recursos, ya que estas relaciones se presentarán tras definirlos.

### **3.2.1.2 Subproceso**

En la definición de un proceso pueden existir pasos complejos que no se puedan representar mediante actividades, puesto que éstas se consideran atómicas. En este caso, cada paso complejo se representa como un subproceso. Un subproceso es un proceso que forma parte de otro proceso y su definición coincide con la del *Proceso*, con la particularidad de que siempre existirá un proceso de nivel más alto al que pertenezca.

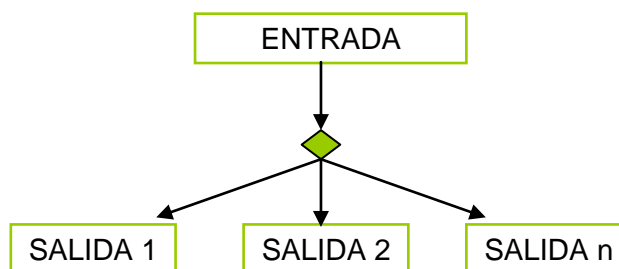
Un flujo de trabajo puede contener uno o más subprocesos como se definan. Los subprocesos son conocidos también como submapas, éstos a su vez pueden ser tan complejos que en muchas ocasiones puede surgir que un submapa sea invocado por otro submapa. Este concepto introduce modularidad en la definición de un flujo de trabajo, facilitando el modelado de procesos complejos.

### 3.2.1.3 Condiciones de Transición

En la definición de un proceso se pueden añadir condiciones de transición que permiten establecer distintas alternativas o caminos a seguir, según se cumplan o no una serie de requisitos. Estas condiciones de transición pueden ser de dos tipos: de bifurcación y de unión. A continuación se describe cada una de ellas.

**Condiciones de bifurcación:** Son puntos de bifurcación dentro del proceso, a partir de los cuales se pueden iniciar en paralelo una o más tareas. Se caracterizan por tener un único punto de entrada y varios puntos de salida. Esto implica tener una única tarea como entrada y varias tareas como salida, que definen los distintos caminos a seguir tras la bifurcación como se muestra en la figura 11. Los caminos de salida pueden tener asociadas condiciones sobre los datos que maneja el proceso o sobre su propia ejecución, como por ejemplo, el estado alcanzado por una actividad.

**Figura 11. Condiciones de bifurcación**



El comportamiento general de la condición de bifurcación es el siguiente: cuando finaliza la ejecución de la tarea de entrada, el control llega a la condición de bifurcación; en ese momento, ésta decide a qué caminos de salida se envía el control para que se inicie la ejecución de las correspondientes tareas de salida.

Concretamente, sólo se enviará a aquellos cuya condición asociada se evalúe a cierto. Las condiciones de bifurcación también son conocidas como *Split*. Dentro de las condiciones de bifurcación se distingue entre *bifurcación total* y *bifurcación parcial*. Esta clasificación depende fundamentalmente del número de tareas de salida que pueden iniciar su ejecución tras la bifurcación:

**Bifurcación total:** Se inicia la ejecución en paralelo de todas las tareas de salida asociadas. No hay condiciones asociadas a los caminos de salida. Para estas tareas, se debe cumplir que ninguno de los datos comunes, se podrán modificar en ninguno de los pasos o tareas en paralelo. Si se definen tareas en paralelo que no cumplen esta condición, se debe reevaluar el modelo, ya que no habría coherencia en los datos y los otros pasos tomarían decisiones con información desactualizada. Las tareas candidatas para este tipo de enrutamiento son tareas no excluyentes

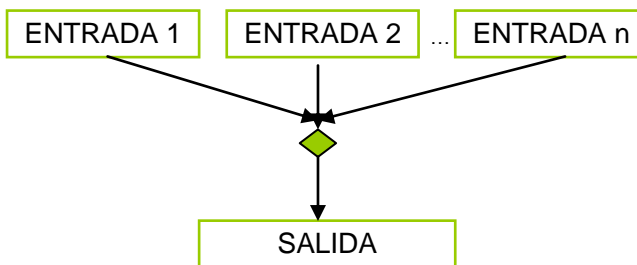
**Bifurcación parcial:** En este caso, sólo se inicia la ejecución de aquellas tareas de salida cuya condición asociada a dicho camino se evalúa a cierto.

En la figura 9 se puede observar la Jerarquía de generalización para las condiciones de bifurcación.

**Condiciones de unión:** Son puntos de reunión, en los que, para ejecutar el siguiente paso dentro del proceso, es necesario que hayan finalizado uno o más de los pasos previos que se están ejecutando de forma concurrente. Las condiciones de unión, por lo tanto, tienen el papel de puntos de sincronización dentro del proceso. Se caracterizan por tener varios puntos de entrada y un único punto como salida, lo que supone tener varias tareas como entrada y una única tarea como salida, como se muestra en la figura 12. Los caminos de entrada pueden tener asociadas condiciones sobre los datos que maneja el proceso o

sobre su propia ejecución, como por ejemplo, el estado alcanzado por una actividad.

**Figura 12. Condiciones de unión**



El comportamiento general de estas condiciones es el siguiente: cada vez que acaba la ejecución de una tarea de entrada, el control llega a la condición de unión, siempre y cuando la condición asociada a dicho camino de entrada ha sido evaluada a cierto; en ese caso, la condición de unión decide si se envía el control al camino de salida para iniciarse la ejecución de la tarea de salida o se espera la finalización de más tareas de entrada, cuya condición asociada sea cierto. Dentro de las condiciones de unión se encuentran *unión total* y *unión parcial*. Esta clasificación depende del número de tareas de entrada que deben haber finalizado antes de que se inicie la ejecución de la tarea de salida.

*Unión total:* Para que se inicie la ejecución de la tarea de salida es necesario que todas las tareas de entrada hayan finalizado. Al igual que en el caso de la bifurcación total, no hay condiciones asociadas a los caminos de salida.

*Unión parcial:* En este caso, no es necesario que haya finalizado la ejecución de todas las tareas de entrada, sino que es posible especificar el número concreto de tareas de entrada que deben haber finalizado para iniciar la ejecución de la tarea



de salida, siempre y cuando las condiciones asociadas sean ciertas.

Las condiciones de transición, tanto de bifurcación como de unión, abren la posibilidad de tener distintos caminos dentro del proceso y que la ejecución de las actividades y subprocesos que lo componen no sea únicamente secuencial. Toda condición de bifurcación debe tener su punto de bifurcación y debe llegar a un punto de unión. En la figura 9 se puede observar la Jerarquía de generalización para las condiciones de unión. Las condiciones de transición se pueden identificar por medio de un nombre o nomenclatura.

Hasta ahora se ha descrito cómo está formado un proceso, sin tener en cuenta ni el flujo de control que conecta las tareas, ni los recursos utilizados por las mismas. Un proceso está formado por al menos una actividad, por cero o más subprocesos y por cero o más condiciones de transición.

Las condiciones de transición se ven reflejadas en el enrutamiento específico de una tarea. Adicional a las condiciones anteriores existen enrutamientos que son basados en *roles o personas*, los cuales, permiten enrutar una tarea dependiendo de la condición definida a un grupo de trabajo o a un individuo en particular, por ejemplo si la aprobación de un producto requiere una autorización se puede enrutar a Pedro Pérez o al rol Autorizador al cual pertenecen varias personas y cualquiera de ellos puede atender la petición. También existen enrutamientos basados en *relaciones*, éstas son basadas en las relaciones existentes en el organigrama de una organización, así por ejemplo los supervisores les reportan a los gerentes, en este caso la asignación de tareas estaría dependiendo de esta relación.

En situaciones reales de negocio, las personas cambian de parecer muy seguido. Al definir el flujo de trabajo se deben analizar los casos en los cuales esto podría ocurrir, de tal manera que el proceso automatizado sea capaz de manejar situaciones en donde se produzcan cambios de parecer, y llevar el proceso a un estado anterior conocido, donde el cambio se pueda realizar.

#### **3.2.1.4 Flujo de Control**

Un flujo de control conecta dos tareas del proceso, especificando la secuencia correcta de ejecución. Siempre se tiene una tarea origen y otra de destino. Las actividades inicial y final, puesto que delimitan el inicio y final de un proceso, sólo admiten un flujo de salida o uno de entrada, respectivamente. El resto de actividades, al igual que los subprocesos, admiten dos flujos: uno de entrada y otro de salida, puesto que siempre van precedidos por tareas. Las tareas que sí pueden tener varios flujos de salida o varios flujos de entrada son las condiciones de transición, según los distintos tipos comentados. En este caso, los flujos de control pueden tener asociadas condiciones para el caso de unión o bifurcación parcial. Cuando se definan las condiciones de bifurcación parcial se debe analizar la prioridad de estas y se recomienda dejar un camino por defecto o sin condición. Se debe garantizar que las tareas siempre tendrán un camino a seguir. Se recomienda para las actividades secuenciales, no especificar una condición para facilitar el diseño e implementación del proceso a automatizar, ya que la condición siempre se debe cumplir.

#### **3.2.2 Recursos**

Los *recursos* se pueden asociar a las tareas que componen un proceso. El término recurso hace referencia tanto a la estructura organizativa como a la infraestructura de tecnologías de la información de la organización para la cual se está

modelando el flujo de trabajo. Concretamente, los recursos pueden ser de tres tipos: *actores*, *datos* y *aplicaciones*. El recurso *actor* representa la participación humana en el flujo de trabajo y corresponde con la dimensión denominada estructura organizativa. La dimensión relativa a la infraestructura de la organización, está representada por los recursos denominados *datos* y *aplicaciones*. Los *datos* representan la información que maneja el flujo de trabajo, mientras que las *aplicaciones* representan los programas invocados como parte de la ejecución del flujo de trabajo.

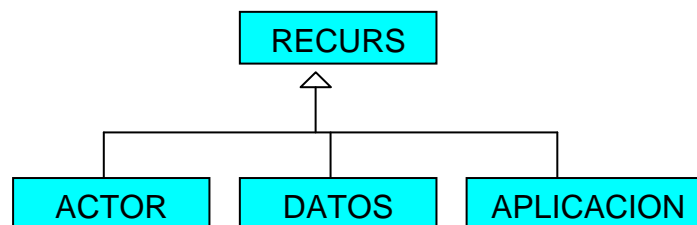
La figura 13 muestra una jerarquía de generalización para los recursos

Las propiedades de los recursos son:

*Identificador – Nombre – Descripción*

Utilizando el operador de especialización definimos las clases que representan a recursos concretos. A continuación se detalla cada uno de los tipos de recursos.

**Figura 13. Jerarquía de Generalización para los recursos**



### **3.2.2.1 Actores y Modelo Organizacional**

Con la noción de *actor* se denota la participación humana en el flujo de trabajo, un proceso puede ser iniciado (o no) por un actor, de la misma forma que una actividad puede ser llevada a cabo por ninguno o varios actores. Los actores que participan en el flujo de trabajo deben ser registrados, es decir, modelados y representados en el proceso que lo define, dando su nombre y una descripción del

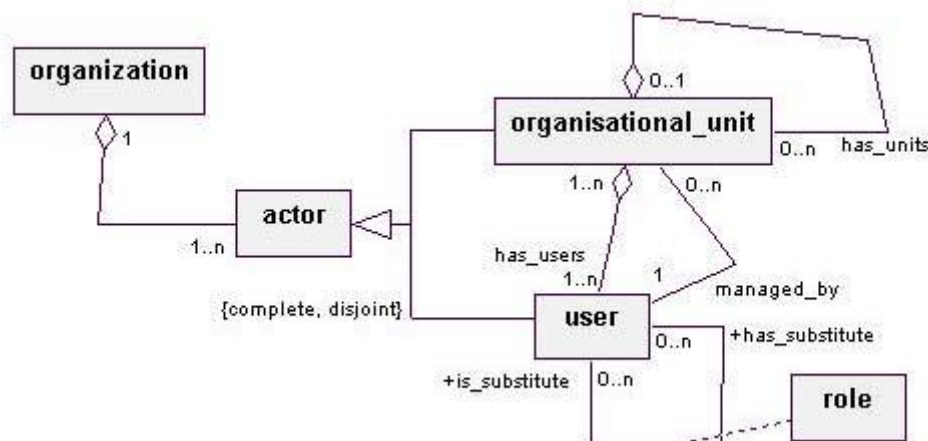
mismo.

El concepto de actor agrupa tanto a cada uno de los *usuarios* que individualmente participan en el proceso como a los posibles grupos de usuarios que puedan participar en el mismo, como departamentos, grupos de trabajo, entre otros. Por ello, se registrará individualmente a cada uno de los usuarios y así como los grupos de usuarios que existan, a los que se llaman de forma genérica *Unidades Organizacionales*. Una unidad organizacional puede estar formada por otras unidades organizacionales. Para registrar a cada unidad organizacional se indicará cuáles son los usuarios que la forman, uno de los cuales será el responsable de la misma. Cuando a una actividad o a un proceso se le asocie como actor una unidad organizacional, cualquier miembro de la misma puede ser el que realmente participe en la ejecución de la actividad o del proceso. En caso de que el actor sea un usuario concreto, sólo éste será el que participe.

Otra noción que tiene que ver con el concepto de actor es la de *Rol*. Los roles son las diferentes actuaciones, representaciones o personificaciones que puede tomar una persona en un proceso. Las personas pueden poseer más de un rol a la vez. Esta noción representa el hecho que un usuario pueda ser sustituido por otro (que pasa a desempeñar la función de aquél). Esto significa que para cada unidad organizacional, que se registre en el sistema, se debe indicar, quién o quienes van a ser los sustitutos de sus miembros, si existen. Modelar la participación humana en el flujo de trabajo, supone tener, en definitiva, un modelo organizacional que permita representar una jerarquía de actores que van a realizar diversas actividades dentro de la organización a la que pertenecen. La figura 14 muestra el modelo organizacional utilizado en literatura de flujos de trabajo [Hol95, Ley00]. En dicho modelo toda organización (representada por la clase *organization*) consta de un conjunto genérico de actores (representado por la clase abstracta *actor*). Una jerarquía de herencia define los distintos tipos de actores que pueden existir. En este caso, existen dos clases que son relevantes. Por una parte, la clase *user* que

representa a los usuarios individuales, y por otra parte, la clase `organisational_unit` que representa a los grupos de usuario, denominados Unidades Organizacionales. La sustitución de un usuario por otro, en la realización de cierta actividad, es modelada mediante una relación recursiva, que tiene asociada una clase (`role`). La jerarquía existente en la organización es modelada mediante tres relaciones. En primer lugar, dos relaciones de composición (denominadas `has_units` y `has_users`) modelan, respectivamente, el hecho de que una unidad organizacional puede estar formada por otras unidades organizacionales, y que una unidad organizacional está formada por usuarios. Los usuarios pueden pertenecer a una o más unidades organizacionales, mientras que una unidad organizacional sólo puede formar parte de otra unidad organizacional de nivel más alto. En segundo lugar, la relación denominada `managed_by` modela el hecho de que toda unidad organizacional tiene, entre los usuarios que la componen, un responsable que es quien la gestiona.

**Figura 14. Modelo organizacional**



Fuente: [Hol95]

### 3.2.2.2 Datos

Los datos representan la información que necesita un flujo de trabajo. En la definición del mismo, los datos van asociados a las tareas que componen un proceso, que los utilizan en su ejecución. Más concretamente, los datos son utilizados por actividades concretas o como parte de la evaluación de las condiciones de transición. En el caso de los subprocesos el tratamiento respecto a los datos es el mismo que para un proceso.

Los datos pueden ser tanto de entrada como de salida y normalmente son persistentes, estando almacenados en lo que genéricamente se puede llamar *Repositorio*. Cuando una actividad comienza su ejecución, consulta del repositorio los datos de entrada que necesite. Durante la ejecución de la actividad o al finalizar la misma, se almacenan en dicho repositorio cualquier dato de salida que se haya podido generar.

Aquellos datos que vayan a ser utilizados en el flujo de trabajo, al igual que ocurre con las aplicaciones, deben ser registrados en el propio sistema, almacenándose la información necesaria para que se pueda acceder a ellos correctamente, dicha información es la siguiente:

*Identificación - Nombre – Descripción - Path*

El Path es la ubicación física o ruta de acceso a los datos, ya sea estructura de directorios local o en otras máquinas. En el caso concreto de los datos, en muchas ocasiones ocurre que el flujo de trabajo no utiliza solamente datos ya definidos, y por lo tanto, existentes en una base de datos, un fichero de texto o cualquier otro formato, sino que es necesario definirlos al definir el flujo de trabajo. Por ejemplo, puede ocurrir que ciertas actividades o condiciones que componen el flujo de trabajo utilicen desde simples variables que almacenen un valor hasta objetos mucho más complejos. Los datos pueden ser archivos, registros en las bases de datos, campos en formularios electrónicos, etc.

### 3.2.2.3 Aplicaciones

Las aplicaciones representan los programas que se invocan como parte de la ejecución de una o más actividades automáticas. Estos programas externos pueden ser aplicaciones informáticas genéricas que se utilizan para la realización de ciertas tareas, o incluso podría tratarse de software ya existente en la organización (*legacy systems*) para ciertas tareas muy concretas.

Las aplicaciones que se integran a los flujos de trabajo se pueden dividir en dos enfoques básicos:

- ⌚ *Integración de procesos*: integración del flujo funcional de procesos entre las aplicaciones.
- ⌚ *Integración de datos*: integración de la información utilizada por las aplicaciones.

Ninguno de los dos enfoques es necesariamente mejor que el otro. Según los requerimientos de la integración se elige el enfoque adecuado para la necesidad

Para que el flujo de trabajo invoque aplicaciones internas se debe tener claro que es muy importante identificar la siguiente información

*Identificador – Nombre – Descripción - Ubicación física – Parámetros – Permisos\_Acceso - Estado*

La ubicación física es la ruta de acceso a la aplicación, ya sea estructura de directorios local o en otras máquinas. Los parámetros son los utilizados para su invocación, los permisos de acceso son los necesarios para poderla ejecutar si se necesitan y se debe conocer el estado de una aplicación, es decir, si la aplicación se encuentra en ejecución o ya ha finalizado. Al igual que para un proceso o una

actividad, la información del estado sólo es relevante una vez el flujo de trabajo esté en ejecución.

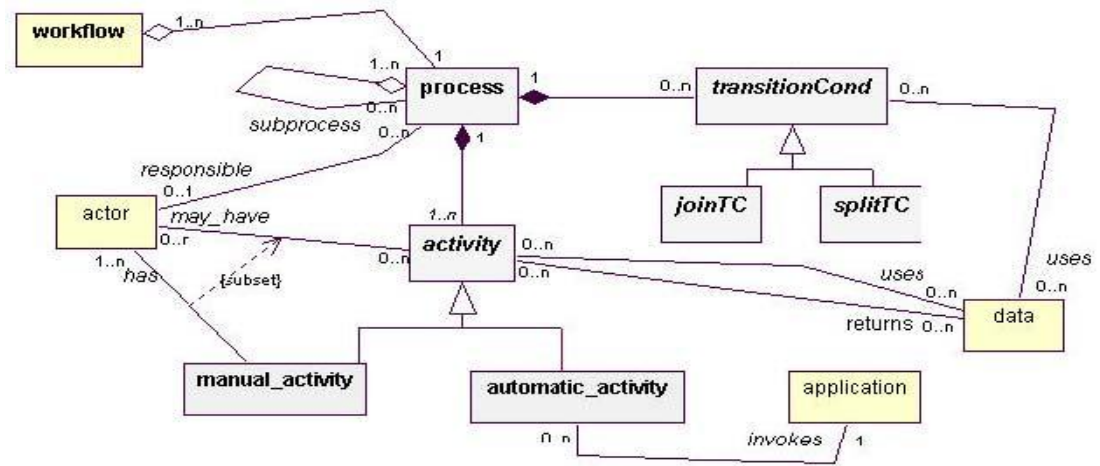
### **3.2.3 Relación entre las actividades y los recursos**

Los datos son utilizados por las actividades y por las condiciones de transición. En la utilización de los datos por parte de las actividades se identifica que si éstos son de entrada, la actividad consulta dichos datos para realizar con ellos cualquier operación, o si son de salida, son generados o actualizados como consecuencia de la ejecución de dicha actividad. Estas dos relaciones definen el flujo de datos dentro del flujo de trabajo. En cuanto a las condiciones de transición, éstas también pueden utilizar los datos como parte de las condiciones que se deben evaluar para determinar por cual ruta o camino continúa el flujo. En este caso, los datos siempre son de entrada puesto que se consultan para evaluar las condiciones, pero no se actualizan.

Los actores pueden estar relacionados con las actividades y con el propio proceso asociado al flujo de trabajo. La relación entre un actor y un proceso representa que dicho actor es el responsable de dicho proceso; esto implica que cualquier incidencia que ocurra durante la ejecución del proceso debe ser notificada al actor. Por otra parte, también puede existir una relación entre uno o más actores y una actividad que compone el proceso. Esto significa que dichos actores son responsables de dicha actividad. En este caso se distingue si la actividad es manual o automática. Si la actividad es manual, la relación se redefine, entendiéndose que dicho actor no sólo es el responsable de la actividad, sino que también es quien debe realizarla. Necesariamente toda actividad manual debe tener asociado al menos un actor. La figura 15 indica las relaciones entre los componentes del flujo de trabajo.



Figura 15. Relaciones entre los componentes del flujo de trabajo



Fuente: [Hol95]

## **CAPITULO 4. PRINCIPALES METODOLOGÍAS QUE PERMITEN EL MODELAMIENTO WORKFLOW DENTRO DE LAS ORGANIZACIONES**

### **4.1 QUÈ ES UNA METODOLOGÍA DE DESARROLLO DE SOFTWARE**

La industria del software, con el correr de los años, ha visto cómo la complejidad en el desarrollo de sistemas ha aumentado de manera dramática. Esta complejidad no solo se presenta en el aspecto tecnológico de un proyecto, también se evidencia en las labores de dirección y control del mismo. Por lo tanto el desarrollo de sistemas se ha convertido en una actividad urgida de estándares y metodologías de control que permitan alcanzar el objetivo que se busca cuando se desarrolla software o producto software.

Un Workflow o flujo de trabajo se puede considerar como un producto software complejo, el cual permite automatizar procesos. Según la Ingeniería del Software, en el desarrollo de todo producto software existe una serie de actividades que deben realizarse en un orden determinado y que abarcan no sólo su producción, sino también su explotación y mantenimiento. Esto se denomina *ciclo de vida* o *proceso de desarrollo del software* [Pre01]. Dentro de este proceso, es necesario tener claros ciertos conceptos que son la base para una automatización adecuada de los procesos, identificar las actividades a seguir para la definición, ejecución y mantenimiento del producto, así como los componentes de un Workflow o Flujo de trabajo.

A continuación se mencionan algunas metodologías existentes para modelar procesos Workflow.

## **4.2 METODOLOGÍAS PARA MODELAR PROCESOS WORKFLOW**

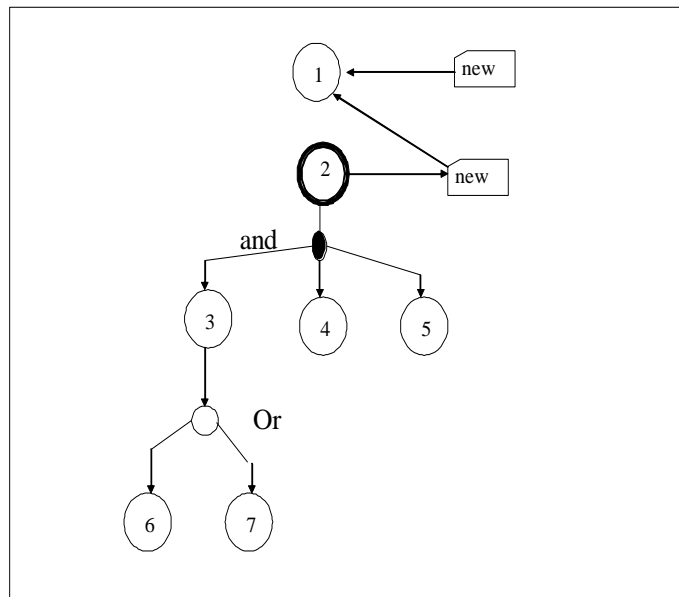
### **4.2.1 Modelo ICN (Information Control Net)**

El ICN es uno de los métodos usados para modelar Workflow, desarrollado por Xerox, el cual, permite representar procesos de trabajo de una organización bajo la forma de un grafo de actividades. ICN visualiza a la organización como un conjunto de procedimientos interrelacionados.

Un procedimiento es un conjunto de actividades ejecutadas por usuarios o grupos de usuarios en un orden específico, relacionadas por restricciones de precedencia, como por ejemplo, el proceso de orden de compra. Cada uno de estos procedimientos se caracteriza por medio de un grafo que describe las actividades que se realizan dentro de un área de trabajo.

Las actividades son unidades de trabajo que se realizan en forma natural, automática o ambas. Ellas reciben o producen información almacenada en diversas formas, como orden de compra, emisión de una factura, aprobación de un crédito, entre otros. Las actividades pueden ocurrir en paralelo durante un procedimiento, o puede existir una relación entre dos o más actividades. La figura 16 muestra como son realizadas las actividades 3, 4 y 5 en forma paralela, después de haberse cumplido la actividad 2.

**Figura 16. Grafo de actividades ICN**



Fuente: [SM01]

Desde el punto de vista de la notación, las actividades son representadas como círculos simples en el caso que los responsables sean usuarios definidos, o círculos dobles en caso de que los responsables sean grupos de usuarios o roles genéricos. Los flujos de control se representan con flechas orientadas que relacionan actividades (relación de precedencia).

Los flujos de información se representan mediante flechas orientadas que unen actividades con repositorios de información (archivos, documentos, mensajes). En este segundo caso, los flujos pueden ser de escritura o de lectura. En otras palabras una flecha entre dos actividades siempre representa un flujo de control.

Los documentos o carpetas de trabajo se mueven entre actividades pasando por depósitos intermedios, puede ser secuencialmente o en paralelo: para esto se definen dos tipos de puntos de sincronización entre actividades:

***Manejo de actividades en paralelo:***

- ⌚ **AND FUENTE:** Bifurcación del flujo tal que las actividades que suceden un “And Fuente” son ejecutadas en paralelo.
- ⌚ **AND SINCRONIA:** Punto de sincronización del flujo; todas las actividades que preceden o convergen en un “And Sincronía” deben haber terminado para seguir con la actividad siguiente.

***Manejo de actividades que dependen de una decisión:***

- ⌚ **OR FUENTE:** Bifurcación condicional tal que una o un subconjunto de las posibles actividades que suceden un “Or Fuente” van a ser ejecutadas.
- ⌚ **OR SINCRONIA:** Punto de sincronización a diferencia del AND la actividad que suceden un “Or Sincronía” es ejecutada si alguna de las tareas que la preceden terminó.  
A cada “OR FUENTE” debe corresponder un OR “SINCRONÍA” para realizar la sincronización de las actividades dentro del grafo.

#### **4.2.2 Redes de Petri**

Las Redes de Petri se han usado tradicionalmente para el modelado de sistemas, entre las técnicas de modelado de sistemas, las Redes de Petri son quizás las que han recibido la mayor atención como candidato potencial para el modelado de procesos de negocio. Las Redes de Petri son representaciones matemáticas / gráficas de sistemas, cuya finalidad es asistir el análisis de la estructura y el

comportamiento dinámico de los sistemas modelados, especialmente sistemas con componentes que interactúan concurrentemente.

Los autores Leymann y Altenhuber han reconocido que las Redes de Petri no son lo suficientemente manejables para ser útiles en el modelado de los procesos de negocio de alto nivel.

#### **4.2.3 Workflow Management Technology**

La administración de procesos de negocio ha sido parte de la cultura IBM desde 1985. Sin embargo, el análisis y rediseño de procesos tomó un nuevo rumbo al enfatizar la creación de nuevos mercados para sus productos y servicios, considerando que los clientes tenían un único requerimiento. Este es el nuevo paradigma de los “Workflows” basado en el trabajo de Fernando Flores y Terry Winograd.

En cada “Workflow” se logra diferenciar un Cliente y un Ejecutor. La tecnología hace énfasis en que en cada proceso existe un cliente que espera un resultado (interno o externo de la organización) y un responsable de que el resultado se obtenga a satisfacción del cliente. Es tan importante la figura del cliente que sin él no hay proceso. Por otro lado, si un cliente se declara satisfecho, el proceso ha cumplido su misión.

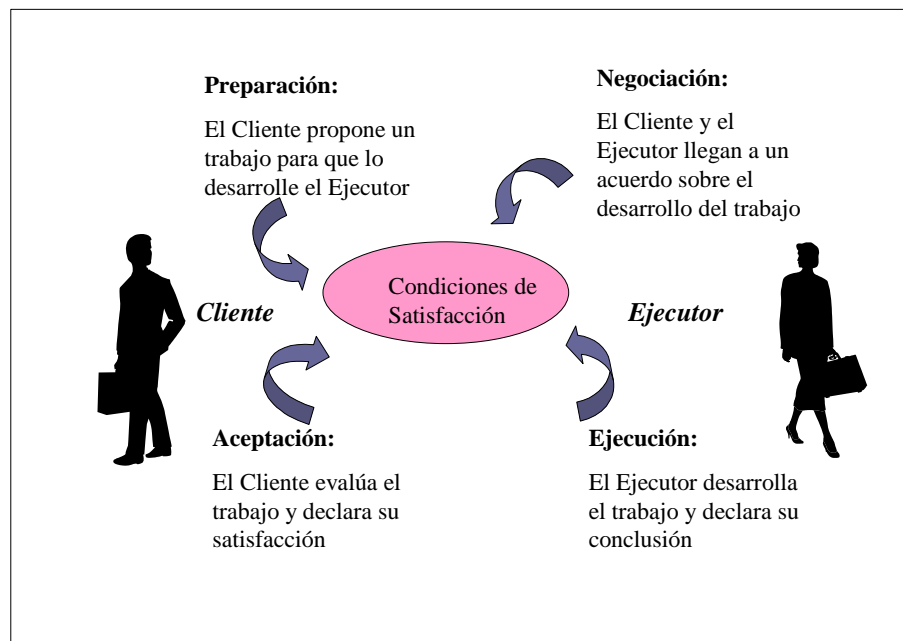
En términos generales, un Workflow puede ser considerado como una herramienta para descubrir porqué ocurren quiebres dentro de la coordinación de la acción humana, como por ejemplo Workflows incompletos, condiciones de satisfacción ambiguas o roles de cliente y ejecutantes que no están del todo claros.

Las interacciones que se producen entre los participantes de un proceso (cliente y ejecutor) se describen por medio de los ciclos de Workflow. Todo ciclo, por tanto, tiene un cliente, ejecutor y condiciones de satisfacción.

El cliente es responsable de establecer las condiciones de satisfacción y evaluar el trabajo reportado. Por otro lado el ejecutor es el que desarrolla el trabajo teniendo como labor reportar su término.

La figura 17 muestra las interacciones de cada uno de los participantes dentro de un Workflow.

**Figura 17. Interacciones Dentro de Workflow**



Fuente: [SM01]

#### **4.2.3.1 Etapas de un Workflow**

Un Workflow tiene cuatro etapas perfectamente definidas, en donde a cada una corresponde un acto lingüístico y dos actores que genéricamente se llaman cliente y Ejecutor.

*Primera etapa:* El cliente realiza una petición o el ejecutor una oferta.

*Segunda etapa:* El cliente y el ejecutor negocian hasta llegar a un acuerdo en torno a las condiciones de satisfacción y al tiempo de terminación. Esto es, se promete realizar la acción en un momento y plazo determinado, o bien se declina finalizado el Workflow.

*Tercera etapa:* El ejecutor realiza el trabajo prometido reportando el cumplimiento de la acción solicitada.

*Cuarta etapa:* El cliente verifica el trabajo realizado por el ejecutor y declara su satisfacción con el trabajo realizado o bien declara no estar satisfecho con lo realizado.

#### **4.2.3.2 Interacciones de un Workflow**

Los tipos de interacciones que pueden ocurrir entre el cliente y el ejecutante son relativamente pequeños. Estos tipos de interacciones dependen de la fase de Workflow en que el cliente y el ejecutante están interactuando como se observa en la Tabla 1.



**Tabla 1. Secuencia de interacciones**

<b>CLIENTE</b>	<i>Petición</i>
<b>EJECUTANTE</b>	Acuerdo
<b>EJECUTANTE</b>	Reporte de trabajo terminado
<b>CLIENTE</b>	Declaración de satisfacción

Cada interacción cambia el estado del Workflow. En la tabla 2 se resumen las posibles interacciones en un Workflow de petición.

**Tabla 2. Interacciones de Workflow de Petición**

<b>FASE DEL CICLO</b>	<b>DESCRIPCIÓN</b>
<b>Preparación</b>	El cliente o el ejecutante propone la ejecución del trabajo
<b>Negociación</b>	El cliente y el ejecutante llegan a un acuerdo en el trabajo a desarrollar
<b>Ejecución</b>	El ejecutante lleva a cabo el trabajo y reporta el término de éste
<b>Aceptación</b>	El cliente revisa el trabajo y declara su satisfacción o rechazo

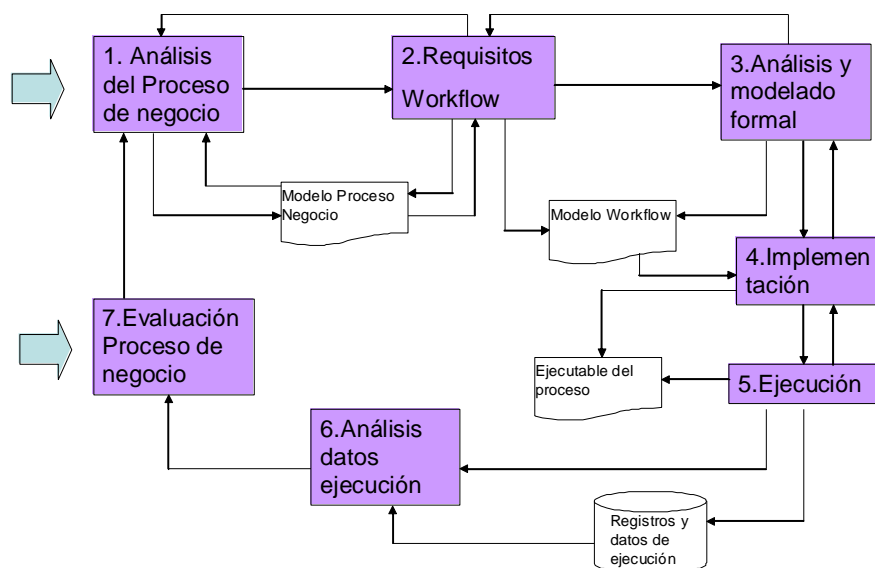
Estas interacciones dan la flexibilidad para acomodarse a un amplio rango de procesos normales y excepcionales. El conjunto de interacciones provee una descripción de cómo las personas se pueden comunicar entre ellas en un trabajo.

## CAPITULO 5. MODELO GENÉRICO PARA LA AUTOMATIZACIÓN DE PROCESOS WORKFLOW

La definición de un modelo de proceso o ciclo de vida a seguir en el desarrollo de la automatización de procesos Workflow conlleva a identificar las actividades a realizar, su orden de ejecución, las entradas y salidas de cada una de ellas. Aunque en la literatura Workflow no existen propuestas explícitas de procesos de desarrollo, se asume con normalidad que existen dos actividades básicas: *la definición del flujo de trabajo y la ejecución del mismo*.

La figura 18 muestra actividades genéricas del ciclo de vida de un proceso Workflow, en las cuales nos centraremos para destacar conceptos y actividades importantes que se deben tener en cuenta al momento de desarrollar un proceso Workflow.

### Figura 18. Actividades del ciclo de Vida en Workflow



<p>1. Análisis del proceso de negocio</p>	<p>Este modelo del proceso de negocio se ha obtenido en una fase de análisis previa al desarrollo del flujo de trabajo, partiendo del supuesto que existía en la organización antes que ésta decidiera automatizar dicho proceso. El análisis de procesos en una organización es útil si necesita mejorar el desempeño de los procesos o se ha presentado una diversificación o reestructuración. El análisis de procesos de negocio mide el desempeño de una organización para identificar áreas de riesgo y oportunidades de mejoramiento. Permite mejorar la productividad y los tiempos de ejecución de los procesos, identificar los riesgos de los procesos y analizar los controles existentes. Incluye un diagnóstico de la organización, análisis de diagramas de flujos, revisión y rediseño de procesos y establecimiento de oportunidades de mejora.</p>
<p>2. Requisitos Workflow</p>	<p>En la Actividad de <i>Requisitos Workflow</i> se identifican los procesos, actividades, recursos utilizados (datos, aplicaciones invocadas y participantes), condiciones de inicio y finalización, y orden de realización de las actividades, de acuerdo con el <i>Modelo del proceso de negocio</i> del que se parte. En la actividad de Requisitos Workflow se pueden utilizar técnicas específicas de Ingeniería de Requisitos como son los Casos de Uso, los cuales permiten capturar los requisitos del proceso y facilitan la obtención de un modelo del mismo, los cuales son complementados en la actividad de <i>Análisis y modelado formal</i>. Esta técnica se explica en el capítulo 7. Existen varias herramientas que apoyan el levantamiento de requisitos a través de los casos de uso como es el caso de REM (Requisite Management) la cual es una herramienta experimental gratuita</p>

	<p>de Gestión de Requisitos diseñada para soportar la fase de Ingeniería de Requisitos de un proyecto de desarrollo software presentada por Amador Durán en septiembre de 2000, también sirve de apoyo para la documentación y descripción de la necesidad del negocio. Con esta herramienta se puede manejar el versionamiento de la documentación, ya que es muy importante tener la trazabilidad de los requisitos iniciales y ver cómo han cambiado a lo largo del desarrollo de un proyecto Workflow.</p>
<p>3. Análisis y modelado formal</p>	<p>Si la actividad 2 no partió de un análisis previo, el <i>Análisis del proceso de negocio</i> y <i>El modelado del flujo de trabajo</i> serían parte de una misma actividad global. Con toda esa información se construye un modelo del proceso a automatizar conocido como <i>Mapa de proceso</i>. Este mapa puede ser construido inicialmente desde antes de los <i>Requisitos Workflow</i> o realizarse en dicha etapa. En la actividad de <i>Análisis y modelado formal</i>, se formaliza dicho mapa de procesos, bien sea si se levantó inicialmente, adaptándolo y refinándolo según la información obtenida en los requisitos o diseñándolo desde cero. Es a partir del mapa de procesos que se permite simular la ejecución del flujo de trabajo o prueba de escritorio, con la finalidad de detectar y corregir errores en el modelado. En las actividades descritas hasta el momento, se interactúa directamente con los usuarios, ya que deben quedar claro todos los requisitos; si en la actividad del <i>Análisis y modelado formal</i> se encuentran dudas o vacíos en la información, se debe aclarar con los usuarios; este proceso será iterativo, repitiéndose el número de veces que sea necesario hasta poder garantizar que la lógica del flujo de</p>

	<p>trabajo modelado está de acuerdo con la del proceso de negocio del que se partió. En el <i>Análisis y modelado formal</i> se debe tener en cuenta todas las actividades que se deben realizar en un proceso de desarrollo de Software como lo son análisis de la arquitectura a utilizar, integraciones con otros sistemas, conectividad, acceso a la base de datos, entre otros. El análisis termina con la validación del modelo propuesto, aprobación de los usuarios y visto bueno del líder del proyecto. Es muy importante definir un repositorio donde se encuentre toda la documentación y donde las personas implicadas en el desarrollo de este proceso tengan acceso. Todos los documentos deben contener su fecha, versión y quien realizó la última modificación y el tipo de modificación realizada como <i>actualización, corrección o adición</i>. Todos los documentos y decisiones a lo largo del proceso deben estar almacenados, para así tener un histórico de todos los cambios o adiciones realizados. La herramienta REM soporta este proceso, ya que durante el análisis pueden realizar modificaciones que afectan la documentación y/o levantamiento de requisitos definido en la actividad anterior.</p>
<p>4. Implementación</p>	<p>En la <i>Implementación</i>, los conceptos de modelado, los cuales se explicarán más adelante, se representan en un formato entendible por una <i>Herramienta Workflow</i>. Es muy importante modelar inicialmente en la Herramienta Workflow el mapa de procesos, ya que es a partir de este, que se van encontrando otros pasos o actividades a realizar, propias del flujo a automatizar y que no se tuvieron en cuenta en el análisis. Este modelamiento que se propone es simple, se trata de partir de lo</p>

	global a lo específico. El resultado de la actividad de Implementación es un Workflow ejecutable del proceso automatizado.
5. Ejecución	<p>La actividad de <i>Ejecución</i>, aparte de proporcionar la funcionalidad que se espera del flujo de trabajo, generará información de ejecución diversa, tal como actividades ejecutadas, duración de cada actividad y recursos utilizados. Esta información se almacenará de forma persistente para su posterior análisis (<i>Registro y datos de ejecución</i>). Es importante aclarar que la información almacenada depende de lo que se halla definido en el <i>Análisis</i> o construido en la <i>Implementación</i>, ya que no se encontrará información de datos que no se definieron, los cuales en la implementación no se tuvieron en cuenta para almacenarlos durante la ejecución. Por eso se recomienda definir todos los datos utilizados por el proceso, asociados al <i>Mapa de procesos</i> y datos adicionales, los cuáles serán informativos y posiblemente se pueda inferir sobre ellos.</p>
6. Análisis datos de ejecución	<p>El <i>Análisis de la información</i> de ejecución del flujo de trabajo o Workflow, debe servir tanto para detectar situaciones anómalas (bloqueos y cuellos de botella), como para poder analizar cualquier otro parámetro de ejecución que se desee (duración media de una actividad, número de veces que se invoca una aplicación externa, actividades que se ejecutaron en un intervalo de tiempo, etc.).</p> <p>La información obtenida en el proceso de <i>Análisis de Workflow</i> puede modificar el modelo del flujo de trabajo construido en la actividad <i>Análisis y modelado formal</i>, si así se cree conveniente. Esto implicaría volver, si es necesario, al proceso inicial de</p>

	Análisis del proceso de Negocio y Levantamiento de Requisitos.
7. Evaluación del proceso de negocio	<p>Las actividades descritas anteriormente, parten del supuesto de que el modelo de proceso de negocio es fijo y lo que se está ejecutando y mejorando es sólo el diseño del flujo de trabajo. Ahora bien, el proceso de desarrollo también incluye mejorar el modelo de proceso de negocio del que se partió.</p> <p>La ejecución del flujo de trabajo hace que quede almacenado persistentemente, no sólo la información descrita en el punto anterior, sino también los datos propios que manejan las distintas actividades involucradas en el flujo de trabajo. Estos datos podrían analizarse a lo largo del tiempo, como una <i>Evaluación del proceso de negocio</i>, cuando el proceso automatizado esté en marcha, en lo que sería un proceso de reingeniería, y chequear si el modelo de proceso de negocio del que partió es el adecuado para los objetivos de la organización en un momento dado; si no es así, habría que revisar dicho modelo.</p> <p>Todos los cambios a realizar en el flujo del proceso deben estar especificados en el modelo de proceso de negocio, previamente analizado dentro de este proceso, también se deben realizar el resto de actividades como el levantamiento de requisitos, análisis y modelado formal. Estos cambios se deben tomar como un desarrollo, el cual, debe contener todas las etapas, se deben documentar en las plantillas correspondientes utilizadas por cada organización y no se deben borrar por ningún motivo los anteriores; lo que se busca es tener la historia del proceso automatizado, para futuras mejoras. Para esto es importante el</p>

	control de versiones de los documentos, donde se va actualizando dicha versión y se especifica el cambio. Los ejecutables del proceso automatizado también sufrirán cambios a lo largo del tiempo, éstos se deben almacenar ordenadamente e identificarlos por medio de una versión. Se debe hacer referencia en alguna parte de la documentación de los requisitos la versión del ejecutable que la contiene, para así identificar fácilmente los cambios en el proceso automatizado.
--	--

## 5.1 ENTORNO DE SOPORTE AL PROCESO DE DESARROLLO

Una vez identificadas actividades a realizar en el proceso de desarrollo de flujos de trabajo, el objetivo es disponer de un entorno que dé soporte a dicho proceso. Concretamente, este entorno debe proporcionar un conjunto de herramientas que permitan realizar las actividades concretas asociadas a cada subproceso o fase descrita anteriormente.

El entorno propuesto interactúa con otros sistemas que van a realizar en todo o parte ciertas actividades dentro del proceso de desarrollo.

El analista debe construir un modelo de flujo de trabajo que represente el proceso de negocio seguido por la organización. Para ello, se apoya de un ***Editor para modelar flujos de trabajo*** que, haciendo uso de un lenguaje gráfico, permite modelar dicho proceso de negocio. Toda la información del modelo gráfico construido queda almacenada con el número de versión en el ***Repositorio*** creado para toda la documentación del proceso. En el capítulo 7 se plantea una estructura de repositorio.



Sin embargo, en muchas ocasiones, especialmente cuando el proceso de negocio es complejo y no totalmente conocido, el analista necesita descubrir dicho proceso antes de modelarlo, siendo necesaria una mayor interacción entre el analista y los miembros de la organización, quienes conocen todo o parte del proceso seguido por la misma. Para dar soporte a esta situación, se ha incluido en el entorno una capa de Ingeniería de Requisitos que permite ayudar al analista a capturar los requisitos del proceso de negocio, expresados mediante casos de uso; un **Editor de casos de uso** permite construir el modelo asociado, a partir del cual se obtiene automáticamente una versión preliminar del modelo de flujo de trabajo. Para implementar el proceso de negocio basado en la información anterior se utiliza una **Herramienta para la automatización de procesos** la cual debe soportar la implementación de todos los requisitos del proceso a automatizar. Para garantizar una buena ejecución del proceso automatizado se deben utilizar **Estructuras de base de datos** que soporten los requisitos a nivel de datos levantados inicialmente. La herramienta para la automatización de procesos luego de modelar y codificar el proceso completo, debe permitir validar la codificación del modelo, **Validador automático**, dicha herramienta también debe permitir generar automática o semiautomáticamente una versión ejecutable del modelo automatizado, a través de lo que se conoce como **Compilador de modelos**. Llegados a este punto, se dispone de un modelo de flujo de trabajo validado que puede ser ejecutado, posteriormente.

Para el análisis del flujo de trabajo, se parte de la premisa que la ejecución del flujo de trabajo por parte de una herramienta Workflow permite crear un registro de ejecución que da cuenta de cierta información relacionada con la ejecución del mismo, tal como instante de inicio, de finalización y estado alcanzado (éxito o fallo) para cada proceso y actividad, instante de asignación y liberación de recursos, entre otros.

Teniendo en cuenta lo anterior, este registro de ejecución se puede analizar para

obtener nueva información acerca del flujo de trabajo que se ha modelado. El análisis comprende tanto la obtención de información puramente estadística como el descubrimiento de conocimiento interesante acerca del proceso, que permita optimizar el flujo de trabajo con base a la información histórica de sus distintas ejecuciones. Dicho análisis se lleva a cabo con la construcción de un **Almacén de datos** que recoge toda la información de ejecución, en el formato adecuado. La estructura del almacén de datos se define en base a los ítems que son interesantes analizar en un flujo de trabajo y que han sido establecidos previamente por la organización.

La tabla 3 resume las herramientas que se utilizan en cada una de las actividades descritas anteriormente.

**Tabla 3. Entorno de soporte al proceso de desarrollo**

<b>Entorno de soporte al proceso de desarrollo</b>	
<b>Herramienta</b>	<b>Actividad</b>
Editor para modelar flujos de trabajo	Requisitos Workflow Análisis y modelado formal
Repositorio	Requisitos Workflow Análisis y modelado formal Implementación
Editor de casos de uso	Requisitos Workflow Análisis y modelado formal
Herramienta para la implementación de proceso	Análisis y modelado formal Implementación
Estructura de Base de datos	Análisis y modelado formal Implementación
Validador automático	Implementación
Compilador de modelos	Implementación
Almacén de datos	Ejecución Análisis de flujos de trabajo

### **PARTE III METODOLOGÍA PARA DESARROLLAR PROCESO WORKFLOW EN UNA COMPAÑÍA DE SEGUROS S.A**

#### **CAPITULO 6. EL PROCESO UNIFICADO DE DESARROLLO DE SOFTWARE**

Cuando el resultado que se busca obtener es un producto software, es necesario hablar de un proceso que permita desarrollar de manera ordenada el producto, para este caso sería un proceso automatizado.

La industria del software, con el correr de los años, ha visto como la complejidad en el desarrollo de sistemas ha aumentado de manera dramática. Esta complejidad no solo se presenta en el aspecto tecnológico de un proyecto, también se evidencia en las labores de dirección y control del mismo. Por lo tanto el desarrollo de sistemas se ha convertido en una actividad urgida de estándares y metodologías de control que permitan alcanzar el objetivo que se busca cuando se desarrolla software.

Un proceso de desarrollo de software describe el cómo construir, desplegar y mantener software. Estas metodologías, técnicas o procesos deben definir quien esta haciendo que, cuando y como lo hace. A lo largo de los años las metodologías de desarrollo han surgido y han muerto. De cada metodología se han aprendido lecciones valiosas que en muchos casos han perdurado en el tiempo convirtiéndose en lo que se conoce como “best practices”. Un proceso de desarrollo debe reunir y presentar estas mejores prácticas.

El proceso unificado de desarrollo de software presenta un conjunto organizado de actividades necesarias para transformar las necesidades o requisitos de un usuario en un producto software. Estas actividades están organizadas en un

“marco de trabajo genérico que puede especializarse para un gran variedad de sistemas software” [Bocja].

Las actividades presentadas en el marco de trabajo del Proceso Unificado se agrupan en disciplinas (anteriormente llamadas flujos de trabajo). Cada una de las disciplinas del Proceso Unificado consiste en una serie de actividades que apuntan a la elaboración de los productos o artefactos importantes para dicha disciplina.

## **6.1 CARACTERÍSTICAS PRINCIPALES**

El proceso unificado de desarrollo de software se caracteriza principalmente por tres puntos claves:

1. *Está dirigido por los casos de uso:* Para el Proceso Unificado de desarrollo de software los casos de uso constituyen el modelo con el cual hilvanan todas las actividades que componen el proyecto. Los casos de uso guían y ordenan el diseño, la implementación y las pruebas del sistema.  
Los desarrolladores implementan los modelos producidos en el diseño del sistema, siempre teniendo en cuenta el cumplimiento del objetivo de lo que se está implementando: Obtener la funcionalidad descrita por el caso de uso. Los ingenieros de prueba deben garantizar que la implementación que realizó el desarrollador cumpla con la funcionalidad descrita en el caso de uso y esperada por el usuario final.
2. *Se centra en la arquitectura:* El Proceso Unificado es centrado en la arquitectura ya que todo el desarrollo del sistema se realiza guiado por las pautas y reglas de la fundamentación del sistema, además, cualquier

cambio en el desarrollo debe ser incluido en la arquitectura. El contar con una arquitectura sólida permite el desarrollo coordinado entre diferentes grupos de trabajo, ya que todos están trabajando bajo las decisiones tomadas en la arquitectura.

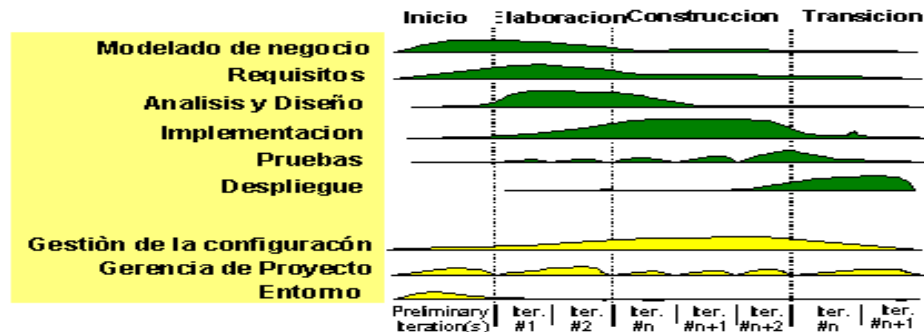
3. *Es iterativo e incremental*: Para algunos autores es la característica más importante del Proceso Unificado [Larman]. La idea central es que el sistema debe ser desarrollado por partes, en periodos de tiempo cortos. De esta manera no se tiene que esperar hasta tener un entendimiento total del sistema para comenzar a obtener resultados, por el contrario, los resultados serán visibles, en la mayoría de los casos, de manera temprana en el proyecto.

En conjunto estas tres características operan de la siguiente manera: En cada iteración se seleccionan los casos de uso que harán parte de la misma. Apoyados en la arquitectura, los analistas de desarrollo toman los casos de uso y los realizan en diagramas de clases, secuencia, etc. (cada proyecto definirá los tipos de diagramas a trabajar). Luego toma los diseños y los implementa, dando como resultado el producto software. Este producto software debe ser verificado para garantizar que cumple con las funcionalidades de los casos de uso. En resumen “la arquitectura proporciona la estructura sobre la cual guiar las iteraciones, mientras que los casos de uso definen los objetivos y dirigen el trabajo de cada iteración” [Bocja]

A medida que se cumple con el ciclo de desarrollo de software (inicio, elaboración, construcción y transición) algunas disciplinas se intensifican mientras otras pierden protagonismo. Según la fase en la cual se encuentre el proyecto se puede mirar la

intensidad en las diferentes disciplinas con el siguiente gráfico. La figura 19 muestra la vista general del proceso Unificado.

**Figura 19. Vista general del proceso unificado**



Fuente: [Marti02]

## 6.2 FASES DEL PROCESO UNIFICADO

*Inicio:* Analizar la viabilidad del sistema. No se trata de un análisis exhaustivo, sino de un análisis que permita tomar la decisión de continuar o no con el proyecto.

*Elaboración:* En esta fase se busca lograr un entendimiento del sistema que se pueda planificar la fase de construcción con una aceptable precisión. De esta fase debe salir una arquitectura coherente que permita la construcción del sistema.

*Construcción:* El objetivo de esta fase es entregar un producto funcional que pueda ser probado. No necesariamente tiene que ser el sistema completo.

*Transición:* El objetivo de esta fase es el de dejar operativo el sistema en el entorno de los usuarios. Esto incluye hacer las correcciones necesarias al producto.

## **CAPITULO 7. TECNICAS UTILIZADAS PARA AUTOMATIZAR PROCESOS WORKFLOW.**

Este capítulo presenta algunas técnicas utilizadas que ayudan al desarrollo de la automatización de procesos Workflow. Algunas de ellas son actividades que se realizan en el proceso de desarrollo de un producto software, pero se hace énfasis en ellas o se mencionan ya que son muy importantes para llevar a cabo esta automatización. También se definen plantillas que son de gran ayuda, las cuáles permiten una mayor organización, búsqueda de información rápida, como por ejemplo cuando se identifica una falla en el proceso. En este capítulo no se describe el proceso de desarrollo el cual es el fin de este trabajo, solo se plantean técnicas o ayudas que se utilizarán en la tercera parte.

### **7.1 DESCRIPCIÓN DEL SISTEMA ACTUAL**

Es la descripción del sistema actual. Para describir el sistema actual, puede utilizarse cualquier técnica que se considere oportuna, por ejemplo, las descritas en [Laguna99] o [Booch99].

### **7.2 OBTENER INFORMACIÓN SOBRE EL DOMINIO DEL PROCESO A AUTOMATIZAR**

Antes de iniciar el levantamiento de requisitos con los usuarios, es fundamental conocer el dominio del problema y los contextos organizacional y operacional, es decir, la situación actual.

Enfrentarse a un desarrollo sin conocer las características principales ni el vocabulario propio de su dominio, suele provocar que el producto final no sea el esperado por los usuarios. Por otro lado, mantener reuniones con los usuarios sin conocer las características de su actividad hará que probablemente no se entiendan sus necesidades y que su confianza inicial hacia el desarrollo, se vea deteriorada enormemente. Esta tarea o actividad es opcional, ya que puede que el equipo de desarrollo tenga experiencia en el dominio del problema y la operatividad del proceso sea conocido.

### 7.3 OBJETIVOS DEL PROCESO A AUTOMATIZAR

Son los objetivos que se esperan alcanzar cuando el proceso se encuentre automatizado y esté en ambiente productivo. Los objetivos se pueden considerar como requisitos de alto nivel [Sawyer1999] de forma que los requisitos propiamente dichos, serían la forma de alcanzar los objetivos.

Estos objetivos se especifican en la plantilla para objetivos que se muestra en la tabla 5.

**Tabla 5. Plantilla de objetivos**

<b>OBJ- &lt;id&gt;</b>	<b>&lt;nombre descriptivo&gt;</b>
<b>Versión</b>	<no de la versión actual> ( <fecha de la versión actual>)
<b>Autores</b>	<autor de la versión actual> ( <organización del autor>)
<b>Fuentes</b>	<fuente de la versión actual> ( <organización de la fuente>)
<b>Descripción</b>	El sistema deberá <objetivo a cumplir por el sistema>
<b>Subobjetivos</b>	OBJ-x <nombre del subobjetivo> ...



<b>Importancia</b>	<i>&lt;importancia del objetivo&gt;</i>
<b>Urgencia</b>	<i>&lt;urgencia del objetivo&gt;</i>
<b>Estado</b>	<i>&lt;estado del objetivo&gt;</i>
<b>Estabilidad</b>	<i>&lt;estabilidad del objetivo&gt;</i>
<b>Comentarios</b>	<i>&lt;comentarios adicionales sobre el objetivo&gt;</i>

A continuación se encuentra el significado de los campos que componen la plantilla de objetivos.

**Identificador y nombre descriptivo:** Cada objetivo debe identificarse por un código único y un nombre descriptivo. Con objeto de conseguir una rápida identificación, los identificadores de los objetivos comienzan con *OBJ*. Los objetivos pueden estar definidos para el proceso en general o para cada paso o actividad del proceso a automatizar.

**Versión:** Para poder gestionar distintas versiones, este campo contiene el número y la fecha de la versión actual del objetivo.

**Autores, Fuentes:** Estos campos contienen el nombre y la organización de los autores, generalmente son los Analistas de Desarrollo y de las fuentes (usuarios), de la versión actual del objetivo, de forma que la rastreabilidad pueda llegar hasta las personas que propusieron la necesidad del negocio.

**Descripción:** Este campo contiene la descripción del objetivo.

**Subobjetivos:** En este campo pueden indicarse los subobjetivos que dependen del objetivo que se está describiendo. Para desarrollo de procesos muy complejos puede que sea necesario establecer una jerarquía de objetivos previa a la identificación de los requisitos. Si esto no es necesario se puede ignorar este

campo.

**Importancia:** Este campo indica la importancia del cumplimiento del objetivo para los usuarios. Se puede asignar un valor numérico o alguna expresión enumerada como *Vital, importante, o quedaría bien*. Si en determinado momento no se establece la importancia se puede indicar que está *Por Determinar (PD)*.

**Urgencia:** Este campo indica la urgencia del cumplimiento del objetivo para los usuarios en el caso de un desarrollo incremental. Se puede asignar un valor numérico o una expresión enumerada como *Inmediatamente, Puede esperar*, llegado al caso que no se halla determinado se puede indicar como *Por Determinar (PD)*.

**Estado:** Este campo indica el estado del objetivo desde el punto de vida de su desarrollo. El objetivo puede estar en *Construcción* si se está elaborando o desarrollando, *pendiente de negociación* si tiene algún conflicto asociado pendiente de solución, *pendiente de verificación* si no tiene ningún conflicto pendiente y está a la espera de verificación o *Validado* si ya ha sido validado por los usuarios.

**Estabilidad:** Este campo indica la estabilidad del objetivo, es decir una estimación de la probabilidad de que pueda sufrir cambios en el futuro. Esta estabilidad puede indicarse mediante un valor numérico o mediante una expresión enumerada como *Alta, Media, Baja* o *Pendiente de Definición*, en el caso de que aún no se haya determinado.

**Comentarios:** Cualquier otra información sobre el objetivo que no encaje en los campos anteriores puede recogerse en este apartado.

## 7.4 ESPECIFICACIÓN DE PARTICIPANTES DEL PROYECTO

Se deben tener claros todos los participantes del proyecto para la automatización de un proceso. Entre los cuales se encuentran los usuarios, líder del proyecto, líder técnico, analistas de desarrollo, analistas técnicos, administradores de las plataformas a utilizar, analistas secundarios (dueños de las aplicaciones a integrar), revisor o auditor. Dentro de las actividades que conforman esta especificación de participantes se encuentra tener claro los nombres y cargos de las personas con la capacidad de tomar decisiones de negocio y hacer definiciones durante el transcurso del proyecto.

En la tabla 6 se encuentra una descripción de algunos participantes o roles del proyecto. Esta plantilla se puede utilizar para identificar los roles del equipo de trabajo de cualquier proyecto.

**Tabla 6. Plantilla de Roles y responsabilidades del equipo de trabajo**

ROL	RESPONSABILIDADES
Líder del Proyecto	<ul style="list-style-type: none"><li>• Velar por el desarrollo de todo el proyecto.</li><li>• Aprobar los requisitos del proyecto para dar inicio de su desarrollo.</li><li>• Tomar decisiones cuando se debe modificar un requisito.</li><li>• Ser el coordinador del cronograma de todo el proyecto.</li><li>• Garantizar que los usuarios quienes vayan a definir requisitos tengan dominio en el tema.</li><li>• Revisar y definir requisitos.</li><li>• Definir reuniones de seguimiento al proyecto y quienes deben ser los participantes de cada una.</li><li>• Estar en contacto con el líder técnico y resolver conflictos si se presentan.</li><li>• Definir el grupo de usuarios quienes definirán requisitos y darán el visto bueno del proceso automatizado a través de pruebas.</li><li>• Conciliar los requisitos con los usuarios si es necesario.</li><li>• Tomar la última decisión sobre el proyecto.</li><li>• Acompañar y realizar el seguimiento a la Planeación, ejecución y verificación de resultados del plan de trabajo del proyecto.</li><li>• Articular las diferentes áreas de la organización implicadas en el proyecto y dirimir en la solución de conflictos para la implementación del plan de trabajo.</li><li>• Representar y defender los intereses de la organización.</li></ul>

Líder Técnico	<ul style="list-style-type: none"> <li>• Realizar el cronograma del proyecto y actualización de este, así como velar por su cumplimiento.</li> <li>• Ser el punto de comunicación entre el líder del proyecto y los integrantes del área de desarrollo y área técnica.</li> <li>• Velar por que los requisitos sean entendibles por los integrantes del equipo, especialmente entre el área de desarrollo, técnica y los usuarios.</li> <li>• Definir reuniones periódicas para analizar y revisar el cumplimiento de las actividades de desarrollo y tecnología.</li> </ul>
Usuarios	<ul style="list-style-type: none"> <li>• Definir requisitos autorizados por el líder del proyecto. El proyecto puede tocar varios frentes del negocio y el líder del proyecto puede asignar a cada frente usuarios responsables para la definición de requisitos.</li> <li>• Tener en cuenta a los usuarios finales para la validación y/o definición de requisitos.</li> <li>• Solucionar dudas a los analistas cuando no se entienda correctamente un requisito.</li> <li>• Velar por que todos los requisitos definidos sean entendidos por los analistas de desarrollo y técnicos.</li> <li>• Validar los casos de uso y prototipos realizados por los analistas de desarrollo.</li> </ul>
Analista Desarrollo	<ul style="list-style-type: none"> <li>• Manejar el ciclo de vida del proceso a automatizar (análisis, diseño, implementación, pruebas, implantación y soporte).</li> <li>• Construir los diagramas de caso de uso.</li> <li>• Realizar el desarrollo del proceso a automatizar cumpliendo con los requisitos planteados por los usuarios líderes.</li> <li>• Proponer mejoras para el buen desarrollo del proceso a automatizar.</li> <li>• Apoyar a los usuarios en el levantamiento de requisitos.</li> <li>• Analizar los requisitos y realizar casos de uso.</li> <li>• Realizar prototipos y validaciones de éstos con los usuarios.</li> <li>• Brindar mantenimiento al proceso automatizado.</li> <li>• Conocer a fondo la herramienta Workflow a utilizar para el desarrollo del proceso.</li> <li>• Utilizar mejores prácticas.</li> <li>• Administrar los datos corporativos y propios del proceso a automatizar.</li> <li>• Garantizar una buena y eficiente integración con las aplicaciones definidas.</li> <li>• Capacitación a los usuarios líderes sobre las soluciones que se desarrollen</li> <li>• Velar por tener actualizada la documentación técnica del proceso.</li> <li>• Realizar manuales de usuario y técnicos</li> <li>• Realizar y actualizar el mapa de procesos</li> </ul> <p>• Se pueden identificar Analistas secundarios, los cuales son los encargados de las aplicaciones con las que se integra el proceso a Automatizar, éstos deben proveer las funcionalidades específicas que el flujo de trabajo requiera, así como la explicación técnica de cómo acceder a dicha aplicación, qué información o parámetros se requieren.</p>
Administrador plataforma Workflow	<ul style="list-style-type: none"> <li>• Proveer la infraestructura tecnológica necesaria para soportar el proceso a automatizar, cumpliendo con estándares tecnológicos y convivencia con la plataforma actual.</li> <li>• Soportar y Mantener disponible la plataforma tecnológica en la que opera la herramienta Workflow.</li> <li>• Administración de incidentes de seguridad en la plataforma Workflow.</li> <li>• Realizar adecuaciones correspondientes en la plataforma para garantizar el buen funcionamiento del proceso.</li> <li>• Velar por tener la versión correcta y actualizada del proceso automatizado en el ambiente productivo.</li> <li>• Velar por la actualización del software base de la herramienta Workflow.</li> </ul>
Analista Técnico	<ul style="list-style-type: none"> <li>• Administrar el licenciamiento que se requiera.</li> <li>• Garantizar la calidad del desarrollo del proceso a automatizar.</li> <li>• Velar por la construcción correcta de los casos de prueba y su ejecución, así como por informar a los analistas de desarrollo las observaciones o errores encontrados en las pruebas.</li> <li>• Velar por el cumplimiento de las siguientes actividades: <ul style="list-style-type: none"> <li>• Velar por la seguridad de la plataforma, proceso automatizado y aplicaciones</li> </ul> </li> </ul>

	integradas. • Administrar las Bases de Datos. • Administrar el respaldo de la información que se encuentra en las bases de datos. • Velar por el cumplimiento del cronograma de pruebas, así como el de implantación.
Revisor o Auditor	• Administrar y definir los usuarios y los diferentes niveles de acceso a la información que se utilice en el proceso automatizado, garantizando así su confidencialidad. • Velar por que la aplicación o proceso cumpla con las políticas de seguridad de la Organización. • Realizar el análisis de vulnerabilidades (Mapa de riesgos, Matriz de vulnerabilidades) y coordinar planes para su solución • A partir de la definición y descripción del proceso, identificar las actividades críticas, entendidas estas como aquellas que hacen franqueable el sistema organizacional y a partir de la medición e identificación de riesgos asociadas a ellas construir las matrices de control (controles detectivos, correctivos y preventivos, automáticos o procedimentales) y auditoria que garanticen razonablemente la buena utilización del proceso automatizado. • Controlar y verificar el cumplimiento de las reglas del negocio establecidas.

## 7.5 REPOSITORIO DE ARTEFACTOS Y DE VERSIONAMIENTO

Es importante tener un directorio compartido donde se encuentre toda la documentación como actas, explicación proceso actual, requisitos o información enviada por los usuarios, casos de uso, casos de prueba, ejecutables y archivos del proceso a automatizar, mapa de procesos, documentación del desarrollo, copia de datos del proceso; todos los documentos e información necesaria.

Esta información debe estar clasificada en subdirectorios y con un nombre nemotécnico alusivo a su contenido. Cada archivo también debe tener su nombre nemotécnico, así como su fecha o versión de realización o actualización.

Todos los miembros del equipo de trabajo deben tener acceso a este directorio. Se puede tener una estructura directorios genérica para todo el proyecto y también se puede tener una estructura por usuario, si se considera necesario. Para este último caso se recomienda que los archivos que se encuentren en la estructura propia de los usuarios, no deben ser archivos para uso común de todos los miembros del equipo de trabajo. Cada documento, plantilla o archivo debe tener un espacio para identificar quién realizó la modificación, fecha en que la

realizó, tipo de modificación como *actualización, corrección o adición*, y descripción del por qué la realizó, la estructura se ilustra en la tabla 7.

**Tabla 7. Estructura de modificaciones**

MODIFICACIONES			
Fecha	Usuario	Versión	Descripción

En las tablas 8 y 9 se da el estándar para los archivos y directorios respectivamente.

**Tabla 8. Estándar de archivos**

Archivo	Estándar Archivo	Ejemplo
Actas	Actas_Seguimiento Actas_Cronograma	Acta01_20060124 Acta01_20060124
Definición_Requisitos	Requisito_X_V_#.#	Eval_Técnica_V_1.0 Eval_Técnica_V_1.1
Casos de uso	Caso_Uso_X_V_#.#	Caso_Uso_inicio_V_2.0 Caso_Uso_inicio_V_2.1
Mapa de procesos	Mapa_Proceso_V_#.#	Mapa_Proceso_V_1.3
Matriz Enrutamiento	Matriz_Enrutamiento_V_#.#	Matriz_Enrutamiento_V_1.7
Flujo	Nombre_Flujo_V_#.#	Nombre_Flujo_V_1.8
Plantilla de pasos	Plantilla_Pasos_V_#.#	Plantilla_Pasos_V_1.10
Plantilla perfilación	Plantilla_Perfilación_V_#.#	Plantilla_Perfilación_V_1.26
Plantilla datos	Plantilla_DF_V_#.#	Plantilla_DF_V_1.3
Plantilla procedimientos	Plantilla_SP_V_#.#	Plantilla_SP_V_1.33
Casos de prueba	Casos_Prueba_Usuarios_V_#.# Casos_Prueba_Tester_V_#.#	Casos_Prueba_Usuarios_V_1.13 Casos_Prueba_Tester_V_1.21
Definición de riesgos y controles	Plantilla_Riesgos_V_#.#	Plantilla_Riesgos_V_1.12
Cronograma	Cronograma_V_#.#	Cronograma_V_1.02

**Tabla 9. Estándar de directorios**

Directorio	Estándar directorio
------------	---------------------

Principal	Nombre_del_proyecto
Actas	Actas_Cronograma
	Actas_Seguimiento
Requisitos	Requisitos
Directorio Desarrollo	Directorio_Desarrollo
	Directorio_Técnica
	Directorio_Usuarios
Cronograma	Cronograma_fecha

Es muy importante tener en cuenta, que la tecnología Workflow garantiza que una tarea ejecutándose en una instancia llegará a su condición de finalización con esa misma instancia.

Se recomienda tener una planilla donde se identifiquen las versiones de los flujos, la tabla 10 muestra la estructura que esta debe tener.

**Tabla 10. Versiones de los flujos**

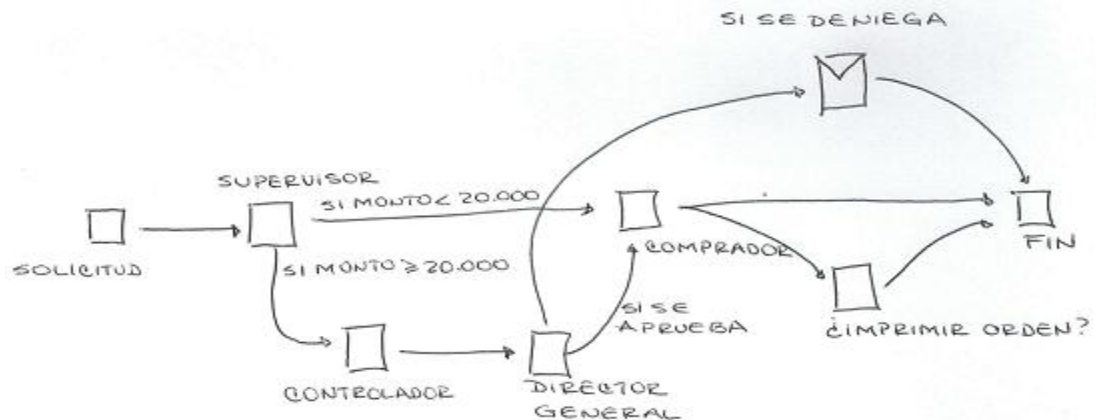
VERSIONES DEL AMBIENTE "X"			
Fecha	Nombre del flujo	Versión	Descripción de la versión

## 7.6 MAPA DE PROCESOS

Un mapa de procesos es la representación gráfica de un proceso. En el mapa deben quedar claras todas las posibles rutas que se puedan tomar, quiénes son los responsables por tarea, qué roles existen, tiempos por tarea etc. Una primera aproximación a la elaboración del mapa habitualmente se expresa en papel y más adelante se utilizan herramientas para la implementación en WF.

La figura 24 es un ejemplo de un mapa modelado en papel.

**Figura 24. Mapa de proceso modelado en papel**



Fuente: [WFGW]

El mapa de procesos modelado en papel, se realiza en el levantamiento de requisito. Dicho mapa puede ir cambiando con el tiempo y el levantamiento de requisitos y mucho mas cuando se realiza la etapa de Análisis y Desarrollo, ya que el mapa de proceso modelado en una herramienta Workflow debe contener los pasos que el usuario no logra detectar y que son requeridos para el desarrollo.

En un mapa de procesos se deben identificar todos los pasos del proceso a automatizar. El mapa de procesos inicialmente sirve para realizar pruebas o un acercamiento al modelo del proceso definido, se recomienda que en las sesiones de levantamiento de requisitos se evalúe con los usuarios dicho mapa, empezando por la ruta del día feliz, así como el análisis de todas las otras posibles rutas, de manera que se logren identificar errores, incoherencias, cuellos de botella que pueden existir dentro de la definición que se está realizando del proceso. Es muy importante que cada uno se imagine el proceso a través de este



Mapa de proceso y las pruebas en papel, por así decirlo, ya que se pretende validar tiempos de ejecución, actividades a realizar en cada paso o actividad.

## 7.7 GLOSARIO DE TÉRMINOS

Se debe definir una lista ordenada alfabéticamente de los términos específicos del dominio del problema, acrónimos y abreviaturas que se utilicen en el proceso de negocio o en el desarrollo del proceso a automatizar, los cuales se considere que su significado deba ser aclarado, para evitar confusiones en el levantamiento de requisitos, en el análisis, desarrollo y pruebas. Cada término debe ir acompañado por su significado.

La estructura de la lista se encuentra en la tabla 11.

**Tabla 11. Estructura glosario de términos**

<b>GLOSARIO</b>	
<b>Término o abreviatura</b>	<b>Significado</b>

## 7.8 LEVANTAMIENTO DE REQUISITOS

Teniendo en cuenta la descripción del proceso actual, objetivos del sistema y la información sobre el dominio del sistema actual se deben preparar y realizar reuniones con los usuarios participantes, con el objeto de obtener sus necesidades y resolver los posibles conflictos que se hayan detectado en iteraciones previas del proceso. Esta tarea es crítica y ha de realizarse con especial cuidado, ya que generalmente el equipo de desarrollo no conoce los detalles específicos de la organización, ni el funcionamiento correcto del sistema actual, por otra parte, los usuarios no saben qué necesita saber el equipo de desarrollo para llevar a cabo su labor. Las técnicas de elicitación más usadas son

entrevistas, lluvia de ideas y casos de uso [Durán2000]. También se recomienda utilizar técnicas de negociación como WinWin.

Es importante identificar los pasos del proceso, funcionalidad de cada uno de ellos, es decir, qué se espera de cada paso. Se debe determinar cada una de las rutas o caminos las cuales identifican cómo van conectados los pasos. Para cada paso se debe también definir cuáles serán las entradas y salidas, cuáles son los datos y/o acciones que debe soportar.

A los usuarios se les debe explicar los términos y datos propios de la tecnología Workflow, para que lo tengan en cuenta al momento de levantar los requisitos.

## **7.9 IDENTIFICACIÓN DE LAS REGLAS DE NEGOCIO**

En los procesos en ocasiones, son necesarias reglas muy complejas y robustas. Todas las reglas de negocio se deben definir ya que pueden o no afectar el enrutamiento. Son definiciones generales del proceso que se determinan como una regla. Muchas de estas contienen información de tiempos de procesamiento actual del proceso (Proceso no automatizado) que servirán como base para futuras referencias. Por ejemplo duración promedio en que se debe demorar una tarea en el paso de aprobación de un producto, si pasados 3 días no se ha realizado la aprobación del producto, se debe notificar al jefe inmediato, si pasados 5 días aún no se ha aprobado el producto se debe enrutar al paso del administrador del producto atendido por el Gerente de ventas. En este ejemplo la regla de duración promedio no afecta enrutamiento, pero la regla que pasado 5 días enrutar al Gerente si afecta el enrutamiento. Se debe definir un alcance para éstas reglas de negocio.

## **7.10 UML COMO LENGUAJE PARA EL ANÁLISIS DE REQUISITOS**

Al momento de enfrentar una realidad, los analistas de desarrollo tienden a realizar diversos diagramas para describir la realidad que se quiere modelar. El tipo de diagrama utilizado depende mucho de la información que se haya adquirido en una etapa de levantamiento de requisitos. En general, los conceptos que se modelan en diversos diagramas utilizados son los mismos. A causa de esto, para un mismo proceso a Automatizar se pueden utilizar los mismos conceptos con distintas representaciones, lo cual dificulta la comunicación con las distintas integraciones que tiene dicho proceso y si son varios analistas analizando y desarrollando puede causar confusión debido que se utilizan diferentes nombres para los mismos conceptos. Es por esto, que se recomienda la utilización de un lenguaje para modelar el proceso que se desea automatizar. UML es un lenguaje de modelado, el cual prescribe un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos, y describe la semántica esencial de lo que estos diagramas y símbolos significan. El lenguaje de modelización es la notación que los métodos usan para expresar el diseño [GxFlow].

Durante el levantamiento de Requisitos como se enunció anteriormente, se recomienda utilizar una herramienta como REM para su captura. Estos finalmente serán documentados a través de casos de uso. Dicha documentación es muy valiosa para la elicitación de requisitos, sin embargo, el proceso de ingeniería requiere el uso de diagramas que permitan el análisis, diseño y construcción de procesos automatizados a partir de dichas especificaciones.

A continuación se describen algunos diagramas que se pueden utilizar para el levantamiento de requisitos y análisis de estos para la automatización de un

proceso. La utilización de estos diagramas depende de cada proceso a automatizar, no es necesario utilizarlos todos, según la experiencia del analista y la descripción del mismo proceso se toma la decisión de que diagramas se utilizarán.

#### **7.10.1 Levantamiento de requisitos con UML**

Un caso de uso describe un conjunto de pasos (denominados *eventos*) que representan la interacción de los elementos externos al proceso a automatizar (denominados *actores*) con el propio proceso automatizado, cuando se ejecuta parte de la funcionalidad contenida en el mismo. Un actor, por lo tanto, describe todo aquello que puede intercambiar información con el proceso, ya sean personas, dispositivos u otros sistemas o procesos. Una vez identificados los distintos casos de uso de un proceso a automatizar, estos junto con las posibles relaciones que existan entre ellos, se denomina *modelo o diagrama de casos de uso*. Entre estas relaciones, las más típicas son: relación de uso o inclusión, relación de extensión y relación de especialización. En otras aproximaciones pueden considerarse otras relaciones, aparte de las mencionadas anteriormente. En [Cas95] se introducen los casos de uso del negocio como aquellos que describen el proceso de negocio de una organización en términos de acciones a realizar y actores que participan, pero siempre bajo la visión del negocio.

De acuerdo con el objetivo que se persigue (modelar el proceso de negocio a automatizar), la construcción de los casos de uso se realiza desde la perspectiva del negocio, representándose los requisitos del negocio y la interacción con sus participantes, denominados actores.

Para obtener mayor información sobre los casos de uso se puede hacer referencia al documento “Metodología para la Elicitación de Requisitos de Sistemas Software” publicado por el Doctor Amador Durán Toro y la Doctora Beatriz Bernádez Jiménez de la Universidad de Sevilla [Durán-Bernádez].

Se recomienda que cada caso de uso haga referencia a un paso del proceso a automatizar, así queda más claro y organizado el levantamiento de los requisitos.

La tabla 12 muestra la plantilla para el levantamiento de requisitos, la cual debe ser diligenciada por el analista de desarrollo luego de haber tenido reuniones con el usuario sobre el levantamiento de requisitos. Los usuarios deben validar esta información; se recomienda que al final de cada reunión de levantamiento de requisitos el analista de desarrollo plasmará lo comprendido en estas plantillas, esto se realiza con el fin de comprender mejor el proceso, y no esperar levantar al final de la toma de requisitos toda la información, sino ir la levantando paso a paso y encontrar puntos de dudas las cuales se deben aclarar en las siguientes reuniones.

**Tabla 12. Plantilla para el levantamiento de requisitos**

<b>FRQ-0001</b>	<b>&lt;nombre descriptivo&gt;</b>
<b>Versión</b>	<no de la versión actual> ( <fecha de la versión actual>)
<b>Autores</b>	<autor de la versión actual> ( <organización del autor>). . .
<b>Fuentes</b>	<Fuente de la versión actual> ( <organización de la fuente>). . .
<b>Dependencias</b>	OBJ-x <nombre del objetivo> FRQ-x <nombre del requisito>
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en el siguiente
<b>Importancia</b>	<importancia del requisito>

<b>Urgencia</b>	<i>&lt;urgencia del requisito&gt;</i>
<b>Estado</b>	<i>&lt;estado del requisito&gt;</i>
<b>Estabilidad</b>	<i>&lt;estabilidad del requisito&gt;</i>
<b>Comentarios</b>	<i>&lt;comentarios adicionales sobre el requisito&gt;</i>

Los campos de versión, autores, fuentes tienen la misma descripción de la plantilla de objetivos aunque referidos a los requisitos. Luego que los usuarios den su visto bueno, el analista debe analizar y detallar mas cada caso de uso, es lo que se conoce como casos de uso extendidos, los cuáles se deben especificar utilizando una plantilla como la que se muestra en la tabla 13.

**Tabla 13. Plantilla para la descripción de casos de uso extendidos**

<b>UC– &lt;id&gt;</b>	<b>&lt;nombre descriptivo&gt;</b>	
<b>Versión</b>	<i>&lt;no de la versión actual&gt; ( &lt;fecha de la versión actual&gt;)</i>	
<b>Autores</b>	<i>&lt;autor de la versión actual&gt; ( &lt;organización del autor&gt;). . .</i>	
<b>Fuentes</b>	<i>&lt;Fuente de la versión actual&gt; ( &lt;organización de la fuente&gt;). . .</i>	
<b>Dependencias</b>	OBJ–x <i>&lt;nombre del objetivo&gt;</i> Rx–y <i>&lt;nombre del requisito&gt;</i>	
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en el siguiente caso de uso { durante la realización de los casos de uso <i>&lt;lista de casos de uso&gt;</i> , cuando <i>&lt;evento de activación&gt;</i> }	
<b>Precondición</b>	<i>&lt;precondición del caso de uso&gt;</i>	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	p1	{El actor <i>&lt;actor&gt;</i> , El sistema} <i>&lt;acción/es realizada/s por actor/sistema&gt;</i>

	p2	Se realiza el caso de uso <i>&lt;caso de uso (RF-x)&gt;</i>
	p3	Si <i>&lt;condición&gt;</i> , {el actor <i>&lt;actor&gt;</i> , el sistema} <i>&lt;acción/es realizadas por actor/sistema&gt;</i> .
	p4	Si <i>&lt;condición&gt;</i> , se realiza el caso de uso <i>&lt;caso de uso (RF-x)&gt;</i> .
<b>Postcondición</b>	<i>&lt;postcondición del caso de uso&gt;</i>	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	Pi	Si <i>&lt;condición de excepción&gt;</i> , {el actor <i>&lt;actor&gt;</i> , el sistema} <i>&lt;acción/es realizada/s por actor/sistema&gt;</i> , a continuación este caso de uso {continúa , termina}
	Pj	Si <i>&lt;condición de excepción&gt;</i> , se realiza el caso de uso <i>&lt;caso de uso (RF-x)&gt;</i> , a continuación este caso de uso {continúa , termina}
	...	...
<b>Rendimiento</b>	<b>Paso</b>	<b>Tiempo máximo</b>
	q	m <i>&lt;unidad de tiempo&gt;</i>
	...	...
<b>Frecuencia esperada</b>	<i>&lt;no de veces&gt; veces / &lt;unidad de tiempo&gt;</i>	
<b>Importancia</b>	<i>&lt;importancia del requisito&gt;</i>	
<b>Urgencia</b>	<i>&lt;urgencia del requisito&gt;</i>	
<b>Estado</b>	<i>&lt;estado del requisito&gt;</i>	
<b>Estabilidad</b>	<i>&lt;estabilidad del requisito&gt;</i>	
<b>Comentarios</b>	<i>&lt;comentarios adicionales sobre el requisito&gt;</i>	

A continuación se encuentra el significado de los campos que componen la plantilla de requisitos.

**Identificador y nombre descriptivo:** Cada caso de uso debe identificarse por un código único y un nombre descriptivo. Con objeto de conseguir una rápida identificación, los identificadores de los casos de uso comienzan con UC. El nombre descriptivo debe ser el nombre del paso o de la actividad identificado en el mapa del proceso.

Los campos de versión, autores, fuentes tienen la misma descripción de la plantilla de objetivos aunque referidos a los requisitos.

**Objetivos asociados:** Este campo debe contener una lista con los objetivos a los que está asociado al caso de uso o el paso, es decir, de los objetivos de los que depende. Esto permitirá conocer qué casos de uso harán que el proceso a automatizar alcance los objetivos propuestos y justifican de esta forma la existencia o propósito del requisito.

**Descripción:** Es la descripción del caso de uso o del paso o actividad. Si el caso de uso es abstracto, debe indicarse los casos de uso en los que se debe realizar, es decir, aquellos desde los que es *incluido* o a los que *extiende*. Si, por el contrario, se trata de un caso de uso concreto, se debe indicar el *evento de activación* que provoca su realización, y en el caso de que sea *incluido* desde, o *extienda* a otros casos de uso, se deberá indicar dichos casos de uso. Debe indicar qué lo que se debe realizar en el paso.

**Precondición:** En este campo se expresa en lenguaje natural las condiciones necesarias para que se pueda realizar el caso de uso. Se indican las posibles rutas, caminos o acciones que hacen que una tarea llegue a este paso.

**Secuencia normal:** Este campo contiene la secuencia normal de interacciones del caso de uso. En cada paso, un actor o el sistema realiza una o mas acciones o



se realiza en otro caso de uso, se debe especificar dicha secuencia y si se pueden realizar varias acciones al tiempo, debe quedara especificado en este campo.

**Postcondición:** En este campo se expresan en lenguaje natural las condiciones que se deben cumplir después de la terminación normal del caso de uso. Se indican las rutas, caminos o acciones que se deben realizar luego de haber terminado dicho paso.

**Excepciones:** Este campo especifica el comportamiento del proceso a automatizar en el caso que se produzca alguna situación excepcional durante la realización de un paso determinado.

**Rendimiento:** En este campo se indica la frecuencia esperada de realización del caso de uso o del paso, que aunque no es realmente un requisito, es una información interesante para los desarrolladores.

**Frecuencia esperada:** En este campo se indica la frecuencia esperada de realización del caso de uso o el número de veces que una tarea puede pasar por este paso, o, para casos excepcionales como por ejemplo una autorización indicar un número aproximado en que puede ocurrir o llegar una tarea a este paso; aunque la frecuencia esperada no es realmente un requisito, es una información interesante para los desarrolladores.

#### **7.10.1.1. Actores**

Los actores de los casos de uso no son requisitos, pero es importante definir cuáles serán los actores que participan en el proceso, identificando su rol. Para una mejor organización y descripción, los actores se pueden definir a través de la plantilla de actores. La tabla 14 muestra esta plantilla.

**Tabla 14. Plantilla de Actores**

<b>ACT– &lt;id&gt;</b>	<b>&lt;nombre descriptivo&gt;</b>
<b>Versión</b>	<no de la versión actual> ( <fecha de la versión actual>)
<b>Autores</b>	<autor de la versión actual> ( <organización del autor>) . .
<b>Fuentes</b>	<fuente de la versión actual> ( <organización de la fuente>) . . .
<b>Descripción</b>	Este actor representa a <rol que representa el actor>
<b>Comentarios</b>	<comentarios adicionales sobre el actor>

A continuación se encuentra el significado de los campos que componen la plantilla de actores.

**Identificador y nombre descriptivo:** Actor debe identificarse por un código único y un nombre descriptivo. Con objeto de conseguir una rápida identificación, los identificadores de los actores comienzan con AC. El nombre puede ser el rol a desempeñar.

Los campos de Versión, Autores, Fuentes tienen la misma descripción de la plantilla de objetivos aunque referidos a los actores.

**Descripción:** Es la descripción del rol o papel que representa el actor respecto al proceso a automatizar.

### 7.10.2 Diagramas UML

Como se describió anteriormente UML presenta varios diagramas los cuáles permiten mayor claridad del proceso a automatizar, visualización de todas las partes que interactúan en dicho proceso y hace mas fácil la implementación de este. Cada proceso a automatizar puede manejar diferentes diagramas, según se requieran. A continuación se presenta una breve descripción de los diagramas mas utilizados UML.

#### 7.10.2.1 Diagramas de casos de uso

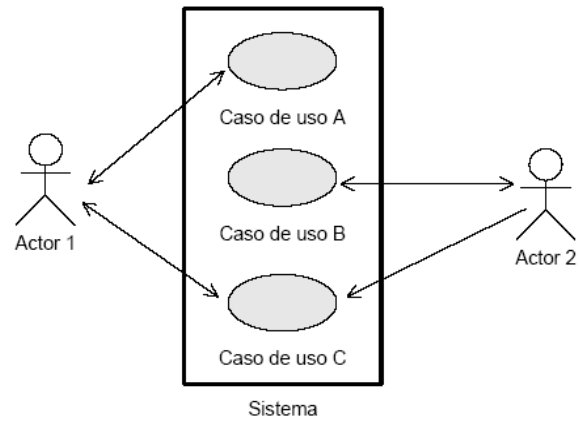
Los casos de uso tienen una representación gráfica denominada *Diagramas de caso de uso*. En estos diagramas, los actores se representan en forma de pequeños monigotes y los casos de uso se representan por elipses contenidas dentro de un rectángulo que representa al sistema. La participación de los actores en los casos de uso se indica por una flecha entre el actor y el caso de uso que apunta en la dirección en la que fluye la información.

Los diagramas de casos de uso sirven para proporcionar una visión global del conjunto de casos de uso de un proceso a automatizar así como de los actores y los casos de uso en los que éstos intervienen.

Las relaciones entre los casos de uso son *includes* y *extends*. Para obtener mayor información sobre el uso de estas relaciones se puede hacer referencia A [Booch99].

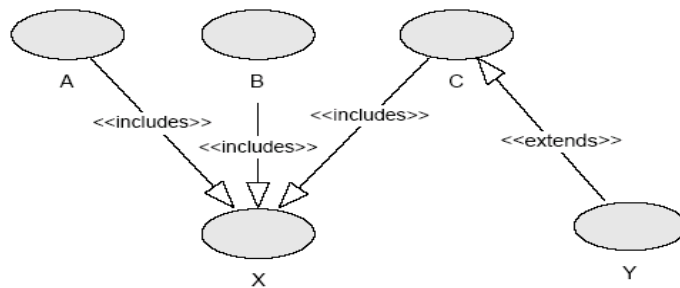
La figura 26 ilustra un diagrama de casos de uso y la figura 27 ilustra la representación gráfica de las relaciones *includes* y *extends*.

**Figura 26. Diagrama de casos de uso**



Fuente: [Durán-Bernádez]

**Figura 27. Representación gráfica de las relaciones de includes y extends**



Fuente: [Durán-Bernádez]

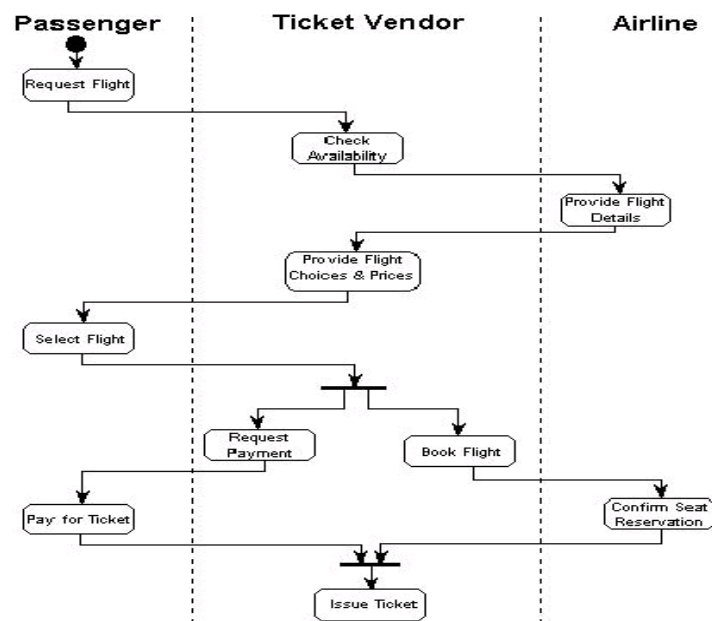
### **7.10.2.2 Diagrama de actividades**

Se utiliza para modelar el proceso en términos técnicos detallando cada uno de los pasos, los flujos de control, de objetos y otras características, como por ejemplo actividades compuestas, acciones, etc. Este será el mapa general del proceso.

Un diagrama de actividades muestra los estados de una actividad, el cual es identificado por un paso atómico en un modelo de procesos, el cual tiene transacciones internas, acciones de entrada y salida y puede abortarse. Las transacciones muestran el camino desde un estado de actividad o acción hacia otro estado de actividad o acción. Una vez que una actividad o acción se completa, entonces el flujo de control pasa inmediatamente al siguiente estado. Si el estado de la actividad tiene una acción de entrada ésta se ejecuta cuando el flujo de control pasa a dicho estado. Si el estado de actividad tiene una acción de salida, ésta se ejecuta cuando el flujo de control sale de dicho estado. En el diagrama de actividades se muestra un modelo secuencial del flujo de control. Los Branching o bifurcaciones condicionales se deben identificar mediante el diagrama de actividades, pues pueden existir saltos o bifurcaciones condicionales para especificar caminos alternativos y secuenciales. Los saltos condicionales tienen un camino de entrada y varios caminos de salida. Las condiciones no deben sobrelaparse, la decisión debe ser determinista, puesto que de lo contrario habría ambigüedades, y completa puesto que de lo contrario el flujo podría detenerse. Se debe identificar las actividades en paralelo utilizando barras de sincronización para especificar puntos de sincronismo. El punto inicial tiene una transición de entrada y varias salidas, estos caminos o rutas continúan en paralelo y son flujos independientes de control. El punto final tiene varias transiciones de entrada y una de salida. Es un punto de reunión o de espera de dos o más flujos de control concurrentes. Los flujos de salida del punto inicial deben corresponderse a los

flujos de entrada del punto final. En la figura 28 se puede observar un ejemplo de un diagrama de actividades.

**Figura 28. Diagrama de actividades**



Fuente: [Cre05]

### 7.10.2.3 Diagrama de Procesos

El diagrama de procesos de negocio, facilita la comprensión del problema desde el punto de vista del negocio. Permite entender el contexto del proceso a automatizar. El diagrama de procesos debe contener como mínimo los siguientes elementos:

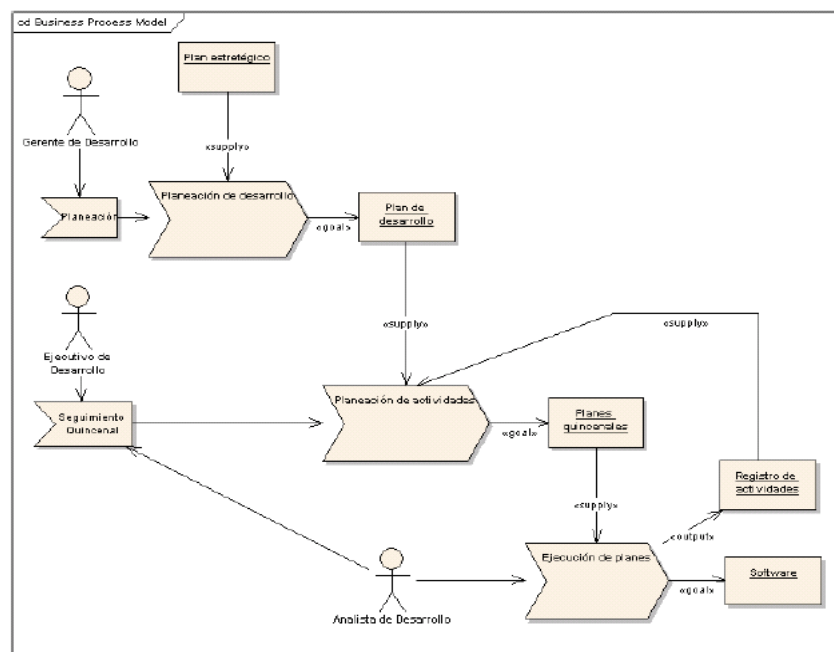
- ⌚ Los actores principales
- ⌚ Los procesos de negocio involucrados en el proceso a automatizar, ya que

se puede tener integraciones con distintos procesos de la Compañía, los cuáles pueden estar o no automatizados.

- ⌚ Las metas del proceso, ya que siempre un proceso debe incluir una meta, la cual es la razón misma del proceso.
- ⌚ Los objetos resultado de cada proceso que interactúa y los insumo necesarios para su ejecución. Los resultados se identifican con el estereotipo <<output>> y los insumos se identifican con el estereotipo <<supply>>.
- ⌚ Los eventos de iniciación de un proceso, ya que cada proceso inicia debido a un evento.

La figura 29 ilustra un ejemplo de un diagrama de procesos.

**Figura 29. Diagrama de procesos**



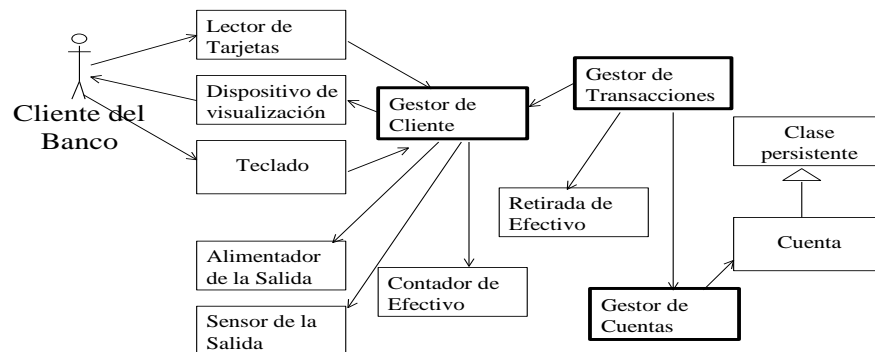
#### 7.10.2.4 Diagramas de clases del dominio

Lo que se pretende con la elaboración de este diagrama es plasmar los conceptos y las relaciones más importantes del proceso a automatizar. De esta manera se inicia el proceso de comprensión de los aspectos estáticos del sistema.

Los diagramas de clases del dominio no incluyen métodos o atributos a menos que se requieran como mecanismo de expresión de alguna característica clave del proceso a automatizar. Además, las relaciones no se elaboran demasiado, la idea es ofrecer un mapa general de los conceptos del negocio que se utilizan en el proceso, en vez de detallar la interpretación en términos de artefactos de software.

La figura 30 ilustra un diagrama de clases del dominio.

**Figura 30. Diagrama de clases del dominio**



#### 7.10.2.5 Diagrama de componentes

El objetivo principal del diagrama de componentes es identificar los componentes



de alto nivel que componen el software que se utilizará en la automatización de un proceso y sus relaciones, además de la forma en que dichos componentes serán implementados por clases concretas.

Una manera de verificar el enfoque del diagrama de componentes consiste en hacerse estas preguntas y poder obtener respuestas concretas:

¿Cuáles son los componentes principales del proceso a automatizar?

¿Cuáles son los componentes externos con los cuáles interactúa el proceso?

¿Cuáles son las interfaces que el proceso automatizado expone a componentes externos?

¿Cuáles son las clases clave que implementan cada uno de los componentes clave del proceso?

¿Están claramente definidas las relaciones y las interfaces entre los componentes del proceso a automatizar?

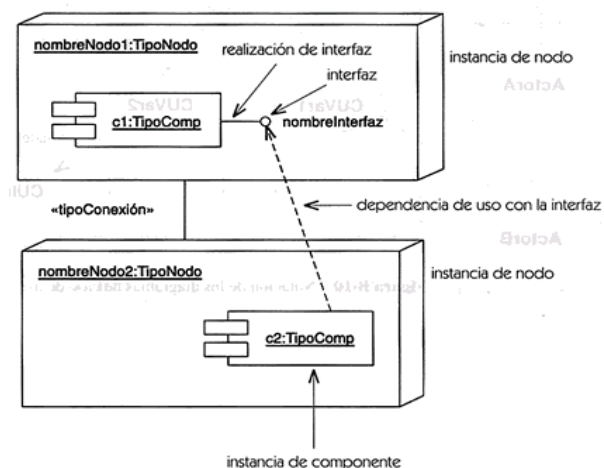
¿Una persona externa al proyecto puede leer rápidamente el diagrama de componentes de alto nivel y entender la composición del proceso a automatizar?

¿Una persona interna del proyecto puede leer el diagrama de componentes detallado y hacerse una idea muy aproximada de la composición y de las relaciones entre los componentes del software?

Para la automatización de procesos Workflow es de gran ayuda este diagrama ya que permite identificar todos los componentes con los que interactúa como pueden ser: capa de presentación, enrutador de tareas, base de datos externas, identificación de cómo será la autenticación a través de una interfaz por medio de los componentes de sincronización, autenticación y servicios, entre otros.

La figura 31 ilustra un diagrama de componentes.

### **Figura 31. Diagrama de componentes**



Fuente: [Cre05]

Para entender un proceso a automatizar es conveniente abordarlo desde diferentes niveles de abstracción iniciando de un nivel macro y finalizando con un nivel detallado. Se sugiere utilizar los siguientes diagramas de componentes.

🕒 *Diagrama de componentes de alto nivel.* Diagrama de componentes de alto nivel en donde se presenten los siguientes componentes:

1. El proceso a automatizar como componente complejo
2. Componentes externos que son utilizados por el proceso
3. Componentes externos que utilizan el proceso automatizado

🕒 *Diagrama de componentes independientes:* Se enfoca al proceso automatizado como un conjunto de componentes que pueden ejecutarse de manera distribuida, es decir, componentes que se podrían ejecutar en procesos independientes. Este diagrama de componentes independientes puede estar compuesto por:

1. Procesos masivos del sistema
2. Demonios
3. Componentes que corren en servidores de aplicaciones

4. Componentes que corren en la base de datos
5. Clientes de o servidores de diferentes protocolos que el sistema deba implementar.

⌚ *Diagrama de componentes detallado:* Un tercer nivel de abstracción es el diagrama de componentes detallado. El objetivo de este diagrama es detallar el corazón del proceso a automatizar al nivel de los componentes que lo conforman. Este diagrama puede estar conformado por:

1. Componentes de interfaz de usuario
2. Componentes de control de flujo
3. Componentes de manejo de flujo de trabajo
4. Componentes de acceso a datos
5. Componentes adaptadores
6. Frameworks utilizados como parte de la implementación del proceso a automatizar,
7. Tablas, archivos y otros artefactos destacables, no es necesario incluir todas las tablas utilizadas en el proceso, la idea es incluir aquellas que cumplen un papel clave, como las de integración, manejo de estados, control de errores, que sean claves para el entendimiento del proceso a automatizar.

⌚ *Diagrama de implementación de componentes:* Se sugiere un diagrama de componentes en el cual se presente de manera detallada pero aproximada la implementación de los componentes en clases concretas. Esta es la guía de implementación del proceso a automatizar, servirá como guía para los desarrolladores y facilitará las labores de mantenimiento. El contenido de este diagrama es:

1. Los componentes detallados en diagramas anteriores
2. Las clases o adaptadores que implementan las interfaces

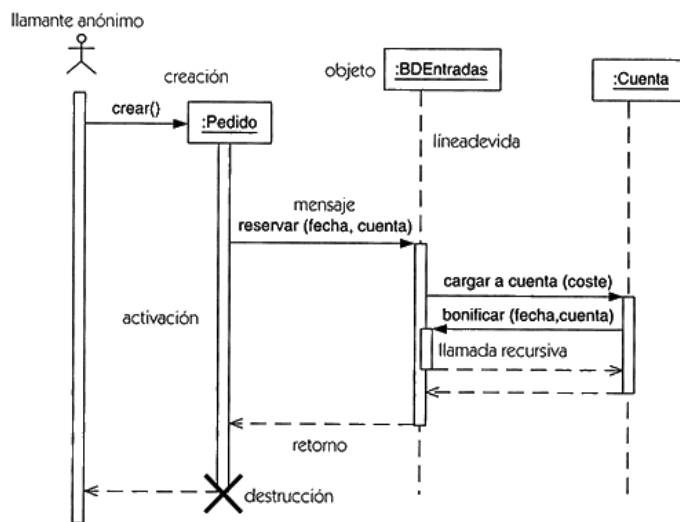
3. Estereotipos que indiquen el tipo de implementación de las interfaces, por ejemplo procedimientos de base de datos, servicio, entre otros.

#### **7.10.2.6 Diagrama de secuencia**

Se recomienda su utilización como herramienta de aclaración, cuando se requiere especificar detalladamente una conversación compleja entre los componentes, la utilización de este tipo de diagramas debe ser orientada más como herramienta de aclaración que como una fuente de información que requiere actualización permanente.

El diagrama de secuencia muestra la secuencia explícita de mensajes y son de mucha ayuda para especificaciones de tiempo real y para escenarios complejos. Cada objeto del diagrama de secuencia viene dado por una barra vertical, el tiempo transcurre de arriba abajo y cuando existe demora entre el envío y la atención se puede indicar usando una línea oblicua. La figura 32 ilustra un diagrama de secuencia.

#### **Figura 32. Diagrama de secuencia**



Fuente: [Cre05]

#### 7.10.2.7 Diagrama de despliegue

Los diagramas de despliegue muestran la disposición física de los distintos nodos o servidores que componen el proceso a automatizar y el reparto de los componentes sobre dichos nodos. La vista de despliegue representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados por enlaces de comunicación. Un nodo es un recurso de ejecución tal como un computador, un dispositivo o memoria. Los estereotipos permiten precisar la naturaleza del equipo como dispositivos, procesadores y memoria.

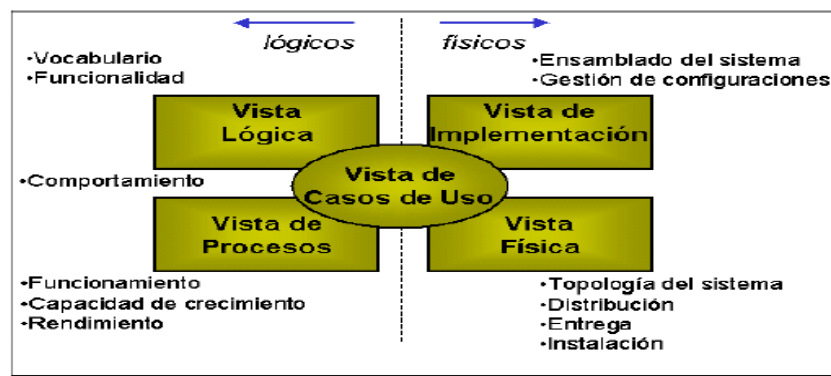
#### 7.10.3 Arquitectura

La arquitectura de software define en términos de abstracciones, composición y descomposición, estilos y estética. Para describir una arquitectura software se puede usar un modelo compuesto por múltiples vistas o perspectivas. El modelo propuesto por Philippe Kruchten está compuesto por 5 vistas, cuatro de ellas

abordan aspectos propios del diseño e implantación del proceso a automatizar y giran en torno a la quinta vista que describe las situaciones mas representativas de la aplicación a través de un conjunto seleccionado de casos de uso.

La figura 33 muestra el modelo de arquitectura 4+1 propuesto por Philippe Kruchten.

**Figura 33. Modelo de arquitectura 4+1**



Fuente: [Grady98]

A continuación se describe el objetivo de cada una de las vistas.

- ⌚ **Vista de casos de uso:** Debido a que las decisiones que fundamenta la arquitectura de software deben estar basadas en las necesidades expuestas en los requisitos, la vista de casos de uso, corresponde a un selecto grupo de casos de uso que sirven como escenarios para describir las otras 4 vistas, en función de los objetivos del negocio. No es necesario incluir todos los casos de uso, solo los que tengan alta complejidad técnica y funcional que por su misma complejidad requieren soluciones técnicas

especiales.

- ⌚ **Vista de procesos:** Esta vista considera algunos requisitos no funcionales como desempeño, disponibilidad y escalabilidad, y está orientada a resolver los posibles problemas de concurrencia, distribución de responsabilidades, integridad del sistema, tolerancia a fallas; también busca solucionar los posibles problemas de concurrencia y sincronización involucrados en el diseño del proceso a automatizar.
- ⌚ **Vista de implementación:** Esta vista se enfoca en el modelo de desarrollo de software de la organización. Esta vista muestra cómo serán ubicados dentro de las capas del proceso a automatizar los diferentes subsistemas que lo componen, esta vista muestra las relaciones entre cada una de ellas.
- ⌚ **Vista física:** Para esta vista se deben considerar principalmente los requerimientos no funcionales del proceso a automatizar, tales como disponibilidad, confiabilidad, desempeño y escalabilidad. Esta vista se ocupa de los aspectos de implantación, es decir, se refiere a los nodos físicos en que se instalarán los componentes del proceso a automatizar y la interacción entre estos, también debe ilustrar la ubicación física de los subsistemas o componentes entregados en la vista de implementación.

## 7.11 MATRIZ DE ENRUTAMIENTO

La matriz de enrutamiento permite identificar las posibles rutas o caminos del proceso a automatizar. Esta matriz debe contener los pasos tanto automáticos como los atendidos por personal humano, en donde se indica la condición de enrutamiento. Se recomienda identificar los pasos automáticos con el estándar

*PA\_Nombre\_Paso*. Donde *PA* indica que es un paso automático y el *nombre\_paso* indica el nombre del paso. A través de la matriz de enrutamiento se pueden identificar las posibles acciones por las cuales una tarea puede llegar a determinado paso. En esta matriz deben estar todos los pasos y condiciones del proceso. La matriz de enrutamiento debe estar siempre actualizada pues es la base para realizar modificaciones al flujo y entendimiento de este. Es insumo de información para el analista desarrollador y los encargados del levantamiento de casos de prueba y realización de los mismos. Se recomienda tener 2 matrices de enrutamiento:

- ⌚ Matriz de enrutamiento para el usuario.
- ⌚ Matriz de enrutamiento exclusivamente para el analista de desarrollo.

En la matriz de enrutamiento de usuario deben estar identificados los pasos de usuario o manuales, así como los pasos automáticos, descritos en los casos de uso. Las condiciones que hacen que un paso se relacione con otro a través de un enrutamiento, serán descritas en palabras de usuario, es decir, se debe colocar el nombre de la acción como la reconoce el usuario, preferiblemente como se encuentra o encontrará en la interfaz de usuario. Por ejemplo en un paso para la Evaluación de un producto, el usuario puede seleccionar dos acciones a realizar, *Pedir autorización* o *Continuar con el proceso*; si el usuario marca *Pedir autorización* la tarea debe enrutarse hacia el paso *Autorizaciones*, si el usuario marca *Continuar con el proceso* la tarea debe enrutarse al paso *Entregar producto*. En la Matriz de enrutamiento de usuario los pasos se deben identificar con el nombre, con el que aparece en la interfaz de usuario como *Pedir autorización* y no con una identificación definida por el analista de desarrollo como *Cola\_1* o *Paso\_1*. Las condiciones de enrutamiento también deben ser identificadas en dicha matriz con la identificación de la interfaz de usuario y no con las condiciones definidas por desarrollo.



La matriz de enrutamiento debe estar definida para cada flujo, es decir, como se vio anteriormente en la parte 1. Pueden existir varios flujos en el proceso automatizado, los cuales se integran o llaman entre sí. Cada flujo tiene su propio enrutamiento, por lo tanto debe tener su propia Matriz de enrutamiento.

Como en el flujo pueden existir subprocessos o submapas, en la matriz de enrutamiento de desarrollo se deben identificar cuáles pasos corresponden a dicho submapa y las relaciones deben estar dadas de paso a paso, de submapa a submapa, de paso a submapa y de submapa a paso, esto con el fin de poder identificar los posibles enrutamientos, ya que un submapa puede comunicarse con otro submapa o paso a través de un enrutamiento. Debe existir una sección en la matriz de enrutamiento la cual indique los pasos que conforman cada submapa.

Para los pasos automáticos se debe identificar qué acción o acciones específicas realiza, como llamado a otra aplicación.

Como se puede dar el caso que las condiciones estén definidas por medio de prioridades, en la matriz de enrutamiento debe existir una sección que indique dichas prioridades, tanto para el usuario como para el desarrollador.

Los pasos paralelos también se pueden identificar en la matriz de enrutamiento, por eso se sugiere que se identifiquen con la prioridad # 0.

Resumiendo lo que debe contener una matriz de enrutamiento es:

- ⌚ Debe existir una matriz de enrutamiento para usuario y para el analista desarrollador.
- ⌚ Cada matriz de enrutamiento debe tener:
  - a. Enrutamiento entre pasos: Indica la relación entre pasos y submapas a través de las condiciones.

- b. Prioridad de condiciones: Indica la prioridad que deben tener las condiciones en un enrutamiento específico.

La tabla 14 muestra un ejemplo de la matriz de usuario. La tabla 15 muestra un ejemplo de la matriz de desarrollo. Estos ejemplos se pueden tomar como plantilla de matriz de usuario y matriz de desarrollo.

**Tabla 14. Ejemplo: Matriz de enrutamiento para el usuario**

MATRIZ DE ENRUTAMIENTO - USUARIOS													
Nombre del proceso: Automatización Proceso de Ventas													
Nombre del flujo: Nombre_Flujo_V_1.8													
Paso salida / Paso llegada	Paso 1		Paso 2		PA_Paso 3		Paso 4		Paso 5		Paso 6		Fin
	Condición	P	Condición	P	Condición	P	Condición	P	Condición	P	Condición	P	
Paso 1	No marca nada	1	Enviar a autorizar	2	Marcar continuar	3							
Paso 2											Marcar continuar	1	
PA_Paso 3							Siempre	0	Siempre	0			
Paso 4											Siempre		
Paso 5											Siempre		
Paso 6													Siempre

**Tabla 15. Ejemplo: Matriz de enrutamiento para el analista**

MATRIZ DE ENRUTAMIENTO - DESARROLLO												
Nombre del proceso: Automatización Proceso de Ventas												
Nombre del flujo: Nombre_Flujo_V_1.8												
Paso salida / Paso llegada	Paso 1		Paso 2		PA_Paso 3		Paso 4		Paso 5		Paso 6	
	Condición	P	Condición	P	Condición	P	Condición	P	Condición	P	Condición	P
Paso 1	not SNAUTORIZAR=True and not SNPASO1=True	1	SNAUTORIZAR=True and SNPASO1=True	2	not SNAUTORIZAR=True and SNPASO1=True	3						
Paso 2											SNPASO2=True	1
PA_Paso 3							SNMOTOR=True	0	SNMOTOR=True	0		

Paso 4																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
--------	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

## 7.12 TIPOS DE DATOS Y DOCUMENTOS

Los datos que se utilicen en el proceso a automatizar se deben identificar, cada dato hace referencia a un campo de base de datos; se debe indicar a qué paso corresponde cada uno y cual es su descripción. Muchos de estos datos pueden ser utilizados por diferentes pasos, por eso es importante definir la utilización de los datos dentro de cada paso, si es de lectura o escritura, así como el tipo.

Muchos pasos pueden hacer uso de documentos, los cuáles están clasificados según políticas y reglas de la organización dependiendo de su uso. Para el usos de documentos se debe especificar cada paso cómo los utilizará.

Existen datos que se utilizarán para analizar su información a futuro, ya que muchos campos son informativos para el funcionamiento y enrutamiento del proceso y otros son fundamentales para la organización. Para estos datos se debe definir si deben permanecer a lo largo de la historia, así como el tipo de almacenamiento que deben tener, bien sea que la herramienta Workflow provee este servicio, o porque en el desarrollo del proceso, se debe implementar una manera para conservar estos datos.

El estándar de nombramiento para los datos, debe ser acorde a los estándares utilizados por la organización.

La tabla 16 indica la matriz de datos que se propone.

**Tabla 16. Matriz de datos**

MATRIZ DE DATOS								
Dato / Utilización	Descripción	Tipo de dato	Ubicación	S/N Dato histórico ?	Paso_1		Paso_2	
					S/N utiliza?	Tipo de Acceso	S/N utiliza?	Tipo de Acceso
Dato_1	Indica ....	Boolean	BD Workflow	S	S	Read	N	—
Dato_2	Nombre ...	String	BD Corporativa	S	S	Read/Write	S	Read
Doc1	Código ....	Integer	Folder 01	N	N	—	S	Read/Write

### 7.13 ROLES

Cómo se describió en la sección 3.2.2.1 los roles son las diferentes actuaciones, representaciones o personificaciones que puede tomar una persona en un proceso o paso en particular. Por esto es importante identificar y describir los roles que se utilizarán en el proceso y luego identificar para cada paso, quienes serán los que atenderán éste, puede darse el caso en que se identifique que un paso puede ser atendido por un usuario específico. Se recomienda que todos los usuarios que vayan a trabajar en el proceso, estén involucrados en algún rol, ya que esto facilita la administración del flujo de trabajo.

La tabla 17 muestra una plantilla para la descripción de los roles. La columna de Integrantes es opcional, ya que cada organización tiene su estructura organizativa, en algunos casos es importante identificar quiénes deben realizar cierto rol.

**Tabla 17. Plantilla de roles**

PLANTILLA DE ROLES					
ROL	Descripción	Integrantes	ROL POR PASOS		
			Paso1	Paso2	Paso2
Rol_1	Administra...	Pedro Pérez Juan Alvarez	X		
Rol_2	Evalúa ....	Mateo Rios		X	
Rol_3	Autoriza ...	Pedro Pérez			X

## 7.14 INTEGRACIONES

Como se identificó en la sección 3.2.2.3 el flujo de trabajo se puede integrar con diversas aplicaciones de la organización. Es importante identificar exactamente en qué parte del flujo se integra cada una de éstas, así como el tipo de integración, por ejemplo si es a nivel de base de datos, si es a través de consumir un servicio ó directamente con la aplicación.

La tabla 18 muestra la plantilla para las integraciones con las distintas aplicaciones, en donde se puede observar cuáles son los pasos en donde se integran las aplicaciones, así como una descripción del tipo de integración que se debe manejar.

**Tabla 18. Plantilla integraciones de aplicaciones**

PLANTILLA INTEGRACIONES CON APLICACIONES			
Paso / Aplicación		Aplicación_1	Aplicación_2
Paso1	Descripción y tipo de integración	A través de un servicio...	Invocar ....
	Identificador	Srv_Mod_Aplica	Aplic_2
	Parámetros	nmrecibo cedula	nombre fecha
	Ubicación Física	//servidor1/aplicaciones	//servidor2/aplicaciones

## 7.15 VALIDACIÓN DE REQUISITOS

Es la etapa donde se asegura que los requisitos describen el producto deseado, validándose con los usuarios. Amador Durán habla de dos técnicas de validación las cuáles son “*Walkthroughs y Prototipos orientados a usuarios*” [Durán 2000]. A continuación se realiza una breve descripción de cada una de ellas.

***Walkthroughs:*** Se presentan a las personas involucradas en el proyecto, usuarios, gestores del proyecto el producto para su aprobación. Esto incluye revisión de diseño y revisión de requisitos. El fin es encontrar conflictos de modo que se puedan plantear alternativas.

***Prototipos orientados a clientes y usuarios:*** Permite que los usuarios tengan una idea más clara del producto, detectar conflictos y requisitos que el usuario no haya manifestado en la elicitación.

Para la automatización de procesos Workflow se recomienda utilizar los prototipos, en los cuáles se pueden identificar las diferentes acciones que se pueden tomar en cada paso. El prototipo debe ir acompañado de la matriz de enrutamiento, mapa de procesos, roles, casos de uso y objetivos del proceso a automatizar para así poder identificar dónde pueden estar posiblemente los conflictos o vacíos en la información o por el contrario certificar que los requisitos están correctos y aceptados. Al validar los requisitos se debe cerrar la versión de estos.

## CONCLUSIONES

- ⌚ Desde el punto de vista teórico se logra destacar la importancia que tiene la tecnología Workflow en el apoyo a los procesos organizacionales, su evolución ha demostrado que a través del tiempo son mas las organizaciones quienes han adoptado dicha tecnología por sus múltiples beneficios y distintas corporaciones han definido y siguen perfeccionando estándares y modelos que permitan cada vez mas la óptima utilización de dicha tecnología.
- ⌚ El desarrollo de procesos Workflow puede verse como un producto software complejo, pues es un desarrollo basado en una tecnología y conceptos propios, que deben pasar por todas las etapas de un procesos de desarrollo de software.
- ⌚ Para la definición de los procesos se hace uso de la Ingeniería de requisitos como una etapa previa al desarrollo, para esto se utilizan casos de uso del negocio para expresar los requisitos, los cuales se obtienen tras un proceso ascendente e iterativo, guiado por la estructura organizacional.

## **TRABAJOS FUTUROS**

- Para un mejor desempeño y actualización de la documentación del proceso por parte del área de desarrollo se puede desarrollar una aplicación que soporte toda la documentación interna que se requiere para implementar un proceso Workfoow, como lo es la matriz de enrutamiento, perfilación de datos y documentos, condiciones del flujo, entre otros.
- Revisar periódicamente la metodología planteada en este trabajo, con el fin de actualizarla y adaptarla según las necesidades que vayan surgiendo en la Compañía y en el medio.
- Definir estereotipos propios para el desarrollo de diagramas UML, con el fin de estandarizar la integración con otras aplicaciones.



## BIBLIOGRAFIA

CYSNEIROS, Luis Marcio. "Requirements Engineering in the Health Care Domain". York University. Reino Unido.

[Alo97] Alonso, G., Agrawal, D., El Abbadi, A., Mohan, C., *Funcionalidad and Limitations of current workflow Management Systems*, IEEE-Expert, 1997.

[Alo98] Alonso, G., Mohan, C., *Workflow Management Systems: The next generation of distributed processing tools*, In Advanced Transaction Models and Architectures, S. Jajodia and L.

[Bocja] BOOCH, Grady and RUMBAUGH, James and JACOBSON, Ivar. El proceso unificado de desarrollo de software. Madrid : Addison Wesley Iberoamericana, 2000. 4 p.

[Boehm94] B. W. Boehm, P. Bose, E. Horowitz, y M.J. Lee. Software Requirements as Negotiated Win Conditions. En *Proceedings of the First International Conference on Requirements Engineering*, 1994. <http://sunset.usc.edu/TechRpts/Papers/NGPM-Requirements93.ps>.

[Booc99] Booch, G., Rumbaugh, J., Jacobson, I., *El Lenguaje de Modelado Unificado. Guía de usuario* Ed Addison-Wesley, 1999.

[Booch99] G. Booch, J. Rumbaugh, y I. Jacobson. *The Unified Modelig Lanaje User Guide*. Addison – Wesley, 19999.

[Cas95] Casati, F., Ceri, S., Pernici, B., Pozzi, G., *Conceptual Modeling of*

*Workflows*. In Proc. of the International Conference of Conceptual Modeling (ER), Gold Coast, Australia, Lecture Notes in Computer Science, Springer Verlag, December 1995.

[Cre05] Consultoría en seguridad Creangel. Disponible en: <http://www.creangel.com/uml/componente.php>

[Cur92] Curtis, B., Kellner, M.I., Over, J., *Process Modelling*. Communications of the ACM, 35(9):75-90, September, 1992.

[DeW91] De Witt, D. et al., *El Manifiesto del Sistema de Base de Datos Orientado al Objeto*, Novática Vol. XVII, núm 91, 1991.

[Durán2000] A. Durán *Un Entorno Metodológico de Ingeniería de Requisitos para Sistemas de Información*. Tesis doctoral, Universidad de Sevilla, 2000

[Durán-Bernádez] DURAN, Amador. BERNANDEZ, B. Metodología para la Elicitación de Requisitos de Sistemas Software (versión 2.2). Disponible en: [http://www.lsi.us.es/~amador/publicaciones/metodologia\\_elicitacion\\_2\\_3.pdf.zip](http://www.lsi.us.es/~amador/publicaciones/metodologia_elicitacion_2_3.pdf.zip)

[Marti02] MARTINEZ Alejandro, MARTINEZ Raúl. Guía a Rational Unified Process. 2002. Disponible en: <http://www.info-ab.uclm.es/asignaturas/42551/trabajosAnteriores/Trabajo-Guia%20RUP.pdf>

[FIL] Visual and Panagon Workflow. FileNet Corporation USA. Disponible en: <http://www.filenet.com/English/Products/index.asp>

[GAR05] GARTNER Group / Meta Group Website [www.gartner.com](http://www.gartner.com)

[Geo95] Georgakopoulos, D., Hornick, M., Sheth, A., *An Overview of Workflow Managenement: from Process Modeling to Workflow Automation Infraestructure*, Distributed and Paralle Databases Vol.3, n.2, April 1995.

[Grady98] Grady Booch, Jame Rumbaugh, Ivar Jacobson, Unified Modeling Lenguaje User Guide, The. Addison Wesley, October 20, 1998.

[Hol95] Hollingsworth, D., *The Workflow Reference Model*, Technical report TC00-1003, WfMC,

[Khoshafian] S. Kohshafian. Introduction to Groupware, Workflow and WorkGroup Computing. Editorial Wiley, 1995. ISBN: 0-471-02946-7.

[Laguna99] M. A. Laguna, J. M. Marqués, y F. J. García. Una Herramienta para la Captura de Requisitos de Usuario. En *Actas de las JISBD'99*, Cáceres, 1999.

[Larman] LARMAN, Craig. Applying uml and patterns. USA; Prentice Hall. 2002. 14 p

[Ley00] Leymam, F., Roller, D., *Production Workflow. Concepts and Techniques*, Prentice Hall, 2000.

[Pre01] Pressman, R.S., *Ingeniería del Software: un enfoque práctico 5ª ed.*, McGraw-Hill, 2001.

[Sad99] Sadiq, W., Orlowska, M., *On Capturing Process Requirements of Worflow Based Business Information Systems*, In Proc. of 3<sup>rd</sup> International

Conference on Business Information Systems (BIS'99), pp. 195-209, Springer-Verlag, Poznan (Poland), April 1999. <http://staff.dstc.edu.au/wasim>

[Sawyer1999] P. Sawyer y G. Montoya. SWEBOK: Software Requirements Engineering Knowledge Area Description. Informe Técnico Versión 0.5, SWEBOK Project, 1999. Disponible en <http://www.swebok.org>.

[She96] Sheth, A. et al., *Report from the NSF Workshop on workflow and Process Automation in Information Systems*. Computer Science Department Technical Report, UGA-CS-TR-96-003, University of Georgia, October 1996. Disponible en: <http://lsdis.cs.uga.edu/activities/NSF-workflow/>

[SM01] Sharp, Alec. McDermott Patrick. Workflow Modeling. Tools for process improvement and application development. Artech House Inc., 2001.

[WFGW] Ultimus. Workflow, Groupware and the rol de Ultimus. July, 2005. Disponible en: <http://www.ultimus1.com>

[WFM98a] Workflow Management Coalition Members, *Workflow Client API Specifications (WAPI)*. Technical report WfMC-TC-1002, WfMC, July, 1998. Disponible en: <http://www.wfmc.org/>.

[WFM98b] Workflow Management Coalition Members, *Workflow Audit Data Specifications*. Technical report WfMC-TC-1015, WfMC, September, 1998. Disponible en: <http://www.wfmc.org/>

[WFM99a] Workflow Management Coalition Members, *Terminology & Glosary*. Technical report WfMC-TC-1011, WfMC, February, 1999. Disponible en: <http://www.wfmc.org/>.

[WFM99b] Workflow Management Coalition Members, *Workflow Standard-Process Definition MetaModel & WPDL*. Technical report WfMC-TC-1016, WfMC, October, 1999. Disponible en: <http://www.wfmc.org/>.

[WFM99c] Workflow Management Coalition Members, *Workflow Standard-Interoperability. Abstract Specification*. Technical report WfMC-TC-1012, WfMC, December, 1999. Disponible en: <http://www.wfmc.org/>.

[WfMC1003] Workflow Management Coalition Members, Reference Model, The Workflow Management Coalition, WfMC- TC-1003, November 29, 1994

[WfMC1012] Workflow Standard - Interoperability Abstract Specification, The Workflow Management Coalition, WfMC – TC –1012, 1.0, 20 Oct. 1996.

[Workflow1011] Terminology & Glossary, The Workflow Management Coalition, WfMC –TC-1011, 3.0, Feb. 1999